

Logiche Direzionali Qualitative

Fondamenti e Motivazioni del Ragionamento Direzionale

Il ragionamento direzionale qualitativo costituisce un pilastro fondamentale della cognizione spaziale umana e dei sistemi artificiali che devono interagire con lo spazio. A differenza delle rappresentazioni metriche che specificano angoli e direzioni attraverso valori numerici precisi, le logiche direzionali qualitative catturano relazioni come "a nord di", "a est di", o "davanti a" in modo simbolico e relazionale.

La necessità di formalizzare il ragionamento direzionale nasce da molteplici considerazioni cognitive e pratiche. Gli esseri umani raramente ragionano sulla direzione utilizzando angoli precisi; piuttosto, impiegano categorie qualitative che dividono lo spazio in regioni discrete basate su sistemi di riferimento naturali o convenzionali. Questa modalità di ragionamento è robusta rispetto a imprecisioni, efficiente dal punto di vista computazionale, e spesso sufficiente per compiti di navigazione, descrizione spaziale, e pianificazione.

Il Cardinal Direction Calculus: Fondamenti Teorici

Il Cardinal Direction Calculus, sviluppato principalmente da Andrew Frank negli anni Novanta, rappresenta uno dei formalismi più influenti per il ragionamento direzionale qualitativo. Questo calcolo si basa sulla divisione dello spazio attorno a un oggetto di riferimento in regioni direzionali corrispondenti ai punti cardinali.

La formulazione classica del Cardinal Direction Calculus considera uno spazio euclideo bidimensionale R^2 e divide questo spazio rispetto a un punto di riferimento o a una regione estesa. Per un punto di riferimento $p = (x_\square, y_\square)$, lo spazio viene partizionato in nove regioni utilizzando due linee ortogonali che passano per p e sono parallele agli assi cardinali.

Le nove regioni sono formalmente definite come segue:

Nord (N): $\{(x, y) \in R^2 : x = x_\square \wedge y > y_\square\}$ **Sud (S):** $\{(x, y) \in R^2 : x = x_\square \wedge y < y_\square\}$ **Est (E):** $\{(x, y) \in R^2 : x > x_\square \wedge y = y_\square\}$ **Ovest (W):** $\{(x, y) \in R^2 : x < x_\square \wedge y = y_\square\}$ **Nord-Est (NE):** $\{(x, y) \in R^2 : x > x_\square \wedge y > y_\square\}$ **Nord-Ovest (NW):** $\{(x, y) \in R^2 : x < x_\square \wedge y > y_\square\}$ **Sud-Est (SE):** $\{(x, y) \in R^2 : x > x_\square \wedge y < y_\square\}$ **Sud-Ovest (SW):** $\{(x, y) \in R^2 : x < x_\square \wedge y < y_\square\}$
Stesso punto (O): $\{(x, y) \in R^2 : x = x_\square \wedge y = y_\square\}$

Questa partizione genera un insieme di nove relazioni direzionali di base, spesso denotato come $DIR_9 = \{N, S, E, W, NE, NW, SE, SW, O\}$.

Relazioni Direzionali tra Regioni Estese

La generalizzazione del Cardinal Direction Calculus a regioni estese piuttosto che punti introduce complessità significative ma anche maggiore espressività. Date due regioni A e B in $\mathbb{R}^2$, la relazione direzionale tra esse non è più una semplice relazione atomica ma può essere una combinazione di più direzioni elementari.

Per determinare la relazione direzionale $DIR(A, B)$, si considera il bounding box minimale di A, cioè il più piccolo rettangolo allineato agli assi che contiene A completamente. Questo bounding box divide lo spazio nelle nove regioni direzionali come descritto precedentemente. La relazione $DIR(A, B)$ è quindi l'insieme delle regioni direzionali che intersecano B.

Formalmente, sia $bbox(A)$ il bounding box di A, e siano $R_N, R_S, R_E, R_W, R_{NE}, R_{NW}, R_{SE}, R_{SW}, R_O$ le nove regioni direzionali definite rispetto a $bbox(A)$. Allora:

$$DIR(A, B) = \{d \in DIR_9 : R_d \cap B \neq \emptyset\}$$

Questo approccio genera $2^9 = 512$ possibili combinazioni direzionali, sebbene non tutte siano realizzabili geometricamente. Un'analisi geometrica dettagliata rivela che solo 218 di queste combinazioni possono effettivamente occorrere per regioni connesse e finite nel piano euclideo.

La Matrice di Proiezione e Rappresentazioni Compatte

Una rappresentazione efficiente per le relazioni direzionali tra regioni estese è la matrice di proiezione 3×3 , introdotta da Goyal e Egenhofer. Questa matrice cattura la struttura topologica e direzionale della relazione tra due regioni.

Data una regione di riferimento A, il suo bounding box viene esteso mediante due linee orizzontali e due linee verticali che passano per i vertici del bounding box, dividendo così l'intero piano in una griglia 3×3 . La matrice di proiezione $M(A, B)$ è una matrice 3×3 dove l'elemento $M[i, j]$ indica se la regione B interseca la cella (i, j) della griglia.

Formalmente, consideriamo il bounding box di A con coordinate minime $(x_{\min}, y_{\min})$ e massime $(x_{\max}, y_{\max})$. Le nove celle della griglia sono definite come:

$$\begin{aligned} C_{00} &= (-\infty, x_{\min}) \times (y_{\max}, +\infty) \text{ (Nord-Ovest)} & C_{01} &= [x_{\min}, x_{\max}] \times (y_{\max}, +\infty) \text{ (Nord)} & C_{02} &= (x_{\max}, +\infty) \times (y_{\max}, +\infty) \text{ (Nord-Est)} \\ C_{10} &= (-\infty, x_{\min}) \times [y_{\min}, y_{\max}] \text{ (Ovest)} & C_{11} &= [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}] \text{ (Centro)} & C_{12} &= (x_{\max}, +\infty) \times [y_{\min}, y_{\max}] \text{ (Est)} \\ C_{20} &= (-\infty, x_{\min}) \times (-\infty, y_{\min}) \text{ (Sud-Ovest)} & C_{21} &= [x_{\min}, x_{\max}] \times (-\infty, y_{\min}) \text{ (Sud)} & C_{22} &= (x_{\max}, +\infty) \times (-\infty, y_{\min}) \text{ (Sud-Est)} \end{aligned}$$

La matrice di proiezione è quindi:

$$M[i, j] = 1 \text{ se } B \cap C_{ij} \neq \emptyset, \text{ altrimenti } M[i, j] = 0$$

Questa rappresentazione compatta permette calcoli efficienti e fornisce una base per il ragionamento sulla composizione di relazioni direzionali.

Cone-Based Reasoning: Approccio Geometrico Avanzato

Il cone-based reasoning rappresenta un'alternativa sofisticata al modello basato su griglie rettangolari. In questo approccio, lo spazio attorno a un oggetto di riferimento viene diviso utilizzando coni direzionali piuttosto che regioni rettangolari.

Un cono direzionale centrato in un punto p con apertura angolare α e direzione principale θ è definito come:

$$\text{Cone}(p, \theta, \alpha) = \{q \in R^2 : \text{angle}(p, q) \in [\theta - \alpha/2, \theta + \alpha/2]\}$$

dove $\text{angle}(p, q)$ denota l'angolo della semiretta da p a q rispetto a un asse di riferimento (tipicamente l'asse x positivo o il nord geografico).

Per il Cardinal Direction Calculus basato su coni, si utilizzano tipicamente quattro coni principali con aperture di 90 gradi, centrati nelle direzioni cardinali:

$$\begin{aligned} \text{Cono_N} &= \text{Cone}(p, 90^\circ, 90^\circ) & \text{Cono_E} &= \text{Cone}(p, 0^\circ, 90^\circ) & \text{Cono_S} &= \text{Cone}(p, 270^\circ, 90^\circ) \\ \text{Cono_W} &= \text{Cone}(p, 180^\circ, 90^\circ) \end{aligned}$$

Un punto q è considerato "a nord di" p se q appartiene al Cono_N centrato in p . Questa definizione è più rigorosa della semplice condizione $y_q > y_p$ perché esclude punti che sono prevalentemente a est o ovest di p .

Il cone-based reasoning si estende naturalmente a regioni estese. Una regione B è in una data direzione rispetto ad A se una porzione significativa di B (secondo qualche misura, tipicamente area o centroide) cade nel cono corrispondente centrato nel centroide di A .

Composizione di Relazioni Direzionali

Un aspetto cruciale delle logiche direzionali è la capacità di comporre relazioni: se A è a nord di B e B è a nord di C , cosa possiamo inferire sulla relazione tra A e C ? La tavola di composizione per le relazioni direzionali specifica il risultato possibile della composizione di due relazioni.

Per le relazioni direzionali atomiche, la composizione non è sempre determinata univocamente. Ad esempio:

$N \circ N = N$ (se A è a nord di B e B è a nord di C , allora A è a nord di C) $N \circ E = \{N, NE, E\}$ (il risultato dipende dalle posizioni specifiche)

La tavola di composizione completa per DIR_9 è una struttura $9 \times 9 \times 2^9$ che specifica per ogni coppia di relazioni direzionali in input l'insieme delle possibili relazioni in output. Questa tavola può essere calcolata geometricamente o derivata algebricamente.

Formalmente, se R_1 e R_2 sono relazioni direzionali, la loro composizione $R_1 \circ R_2$ è definita come:

$(A, C) \in R_1 \circ R_2$ se e solo se esiste B tale che $(A, B) \in R_1$ e $(B, C) \in R_2$

Per regioni estese, la composizione diventa più complessa perché le relazioni sono insiemi di direzioni atomiche. La composizione di due relazioni complesse R_1 e R_2 è data da:

$$R_1 \circ R_2 = \bigcup \{d_1 \circ d_2 : d_1 \in R_1, d_2 \in R_2\}$$

Questa operazione può portare a una rapida crescita dell'incertezza: componendo molte relazioni, il risultato potrebbe includere tutte le nove direzioni possibili, rendendo l'informazione inutile.

Constraint Satisfaction e Consistenza

Il ragionamento direzionale qualitativo può essere formulato come un problema di Constraint Satisfaction Problem (CSP). Un network di vincoli direzionali è una tripla (V, D, C) dove:

- V è un insieme finito di variabili, ciascuna rappresentante un oggetto spaziale
- D è un insieme di domini, dove D_i è l'insieme delle possibili posizioni o regioni per la variabile v_i
- C è un insieme di vincoli, dove ogni vincolo è una relazione direzionale tra due variabili

Un vincolo direzionale tra v_i e v_j è espresso come (v_i, R_{ij}, v_j) dove $R_{ij} \subseteq \text{DIR}$, specifica le relazioni direzionali ammissibili.

Il problema di consistenza chiede: esiste un'assegnazione di regioni alle variabili tale che tutti i vincoli direzionali siano soddisfatti? Questo problema è in generale NP-completo per il Cardinal Direction Calculus con regioni estese.

Tuttavia, tecniche di consistenza locale possono essere applicate per eliminare inconsistenze evidenti. La path-consistency è una forma di consistenza locale particolarmente utile: per ogni tripla di variabili (v_i, v_j, v_k) , la relazione diretta R_{ij} deve essere consistente con la composizione delle relazioni indirette attraverso v_k .

Formalmente, il network è path-consistent se per ogni i, j, k :

$$R_{ij} \subseteq R_{ik} \circ R_{kj}$$

Algoritmi efficienti, come l'algoritmo PC-2, possono imporre la path-consistency in tempo polinomiale, sebbene questo non garantisca la consistenza globale per tutti i calcoli qualitativi direzionali.

Raffinamenti e Granularità Variabile

Il Cardinal Direction Calculus può essere raffinato a diversi livelli di granularità a seconda delle esigenze applicative. Il modello a 9 regioni può essere esteso a 16 regioni incorporando direzioni intermedie:

$DIR_{16} = \{N, NNE, NE, ENE, E, ESE, SE, SSE, S, SSW, SW, WSW, W, WNW, NW, NNW\}$

Questa estensione fornisce una discriminazione più fine delle direzioni ma aumenta la complessità della tavola di composizione che diventa 16×16 .

Un approccio alternativo considera granularità variabili gerarchiche. Al livello più grossolano, lo spazio può essere diviso in quattro quadranti principali. A livelli più fini, ogni quadrante viene ulteriormente suddiviso. Questa struttura gerarchica permette ragionamenti a diversi livelli di dettaglio e facilita l'astrazione e il raffinamento progressivo.

La gerarchia può essere rappresentata formalmente come un albero dove ogni nodo corrisponde a una regione direzionale e i figli di un nodo rappresentano le sottoregioni in cui è diviso. La radice rappresenta l'intero spazio, e le foglie rappresentano le direzioni atomiche al livello di granularità più fine disponibile.

Direzioni Relative e Frame di Riferimento Intrinseci

Una distinzione fondamentale nella cognizione spaziale umana è quella tra direzioni assolute (basate su coordinate globali come i punti cardinali) e direzioni relative (basate su un osservatore o su proprietà intrinseche degli oggetti).

Le direzioni relative includono concetti come "davanti a", "dietro a", "a sinistra di", "a destra di". Queste direzioni dipendono da un frame di riferimento che deve essere specificato esplicitamente. Esistono tre tipi principali di frame di riferimento:

Frame Intrinseco: Basato sulle proprietà geometriche dell'oggetto di riferimento stesso. Ad esempio, "davanti alla macchina" si riferisce alla direzione verso cui punta il fronte della macchina, indipendentemente dalla posizione dell'osservatore.

Frame Relativo: Basato sulla posizione e l'orientamento di un osservatore. "A sinistra della statua" dipende da dove si trova l'osservatore che guarda la statua.

Frame Assoluto: Basato su coordinate globali fisse, come i punti cardinali geografici. "A nord della torre" è invariante rispetto alla posizione dell'osservatore.

La formalizzazione delle direzioni intrinseche richiede di associare a ogni oggetto un sistema di coordinate locale. Per un oggetto A con posizione $p_A = (x_A, y_A)$ e orientamento θ_A (angolo rispetto all'asse x), possiamo definire le direzioni intrinseche mediante una trasformazione di coordinate.

Un punto q è "davanti" ad A secondo il frame intrinseco se, trasformando q nel sistema di coordinate locale di A, la sua coordinata longitudinale è positiva. Formalmente, sia T_A la matrice di trasformazione:

$$T_A = [\cos(\theta_A) \ -\sin(\theta_A) \ -x_A] [\sin(\theta_A) \ \cos(\theta_A) \ -y_A] [0 \ 0 \ 1]$$

Applicando questa trasformazione a un punto q, otteniamo le coordinate q' nel sistema locale. Se $q'_x > 0$, allora q è "davanti" ad A. Analogamente, q è "dietro" se $q'_x < 0$, "a sinistra" se $q'_y > 0$, e "a destra" se $q'_y < 0$.

Trasformazioni tra Frame di Riferimento

Una capacità essenziale per sistemi di ragionamento spaziale sofisticati è la trasformazione tra diversi frame di riferimento. Dato un insieme di relazioni direzionali espresse in un frame, come possiamo convertirle in un altro frame?

Consideriamo la trasformazione da frame intrinseco a frame assoluto. Se sappiamo che B è "davanti" ad A nel frame intrinseco di A, e conosciamo l'orientamento di A rispetto al nord (ad esempio, A è orientato a 45° rispetto al nord), possiamo inferire che B è a nord-est di A nel frame assoluto.

Formalmente, sia $r_intrinsic(A, B)$ la relazione intrinseca tra A e B, e sia $orient(A)$ l'orientamento di A nel sistema assoluto. La trasformazione è data da:

$$r_absolute(A, B) = rotate(r_intrinsic(A, B), orient(A))$$

dove $rotate$ è una funzione che ruota una direzione di un angolo specificato. Per direzioni qualitative, questa funzione mappa insiemi di direzioni in insiemi di direzioni.

La trasformazione inversa, da frame assoluto a frame intrinseco, richiede la conoscenza dell'orientamento dell'oggetto di riferimento:

$$r_intrinsic(A, B) = rotate(r_absolute(A, B), -orient(A))$$

Quando l'orientamento è noto solo qualitativamente (ad esempio, "A è orientato approssimativamente a nord"), la trasformazione introduce incertezza. Se $orient(A) \in \{N, NE, NW\}$, allora $r_absolute(A, B)$ sarà l'unione delle possibili direzioni risultanti per ciascun orientamento.

Logiche Modali Direzionali

La formalizzazione logica del ragionamento direzionale può essere effettuata mediante logiche modali con operatori direzionali. Il linguaggio della logica modale direzionale estende la logica proposizionale con operatori modali per ciascuna direzione:

$$\phi ::= p \mid \neg\phi \mid \phi \wedge \psi \mid \Diamond_d \phi \mid \Box_d \phi$$

dove d varia sull'insieme delle direzioni (ad esempio, $\{N, S, E, W, NE, NW, SE, SW\}$).

L'operatore $\Diamond_d \phi$ è letto "esiste una regione nella direzione d che soddisfa ϕ ". La sua semantica è definita su modelli direzionali $M = (R, DIR, V)$ dove R è un insieme di regioni, $DIR: R \times R \rightarrow P(DIR_s)$ è una funzione che assegna a ogni coppia di regioni l'insieme delle relazioni direzionali che le collega, e V è una funzione di valutazione.

Le condizioni di verità sono:

$$M, r \models \Diamond_d \phi \text{ se e solo se esiste } r' \in R \text{ tale che } d \in DIR(r, r') \text{ e } M, r' \models \phi$$

L'operatore duale $\Box_d \varphi$, definito come $\neg \Diamond_d \neg \varphi$, esprime che tutte le regioni nella direzione d soddisfano φ .

Un sistema assiomatico per la logica modale direzionale include:

Assioma K Direzionale: $\Box_d(\varphi \rightarrow \psi) \rightarrow (\Box_d \varphi \rightarrow \Box_d \psi)$

Assioma di Esclusività: $\Diamond_d \varphi \wedge d \neq d' \rightarrow \neg \Diamond_{d'} \varphi$ (per direzioni atomiche disgiunte)

Assioma di Composizione: $\Diamond_{d_1} \Diamond_{d_2} \varphi \rightarrow \Diamond_{(d_1 \circ d_2)} \varphi$

dove $d_1 \circ d_2$ denota la composizione delle direzioni d_1 e d_2 .

La completezza di questi sistemi rispetto a classi appropriate di modelli direzionali è oggetto di ricerca. La decidibilità della logica modale direzionale è generalmente garantita quando il dominio delle regioni è finito, ma diventa problematica per domini infiniti.

Aspetti Cognitivi e Modelli Mentali

Il ragionamento direzionale qualitativo riflette modalità cognitive umane profondamente radicate. Studi psicologici hanno dimostrato che gli esseri umani preferiscono categorie direzionali discrete e spesso introducono distorsioni sistematiche quando stimano direzioni.

Un fenomeno ben documentato è la "cardinal direction bias": le persone tendono a sovrastimare la prevalenza delle direzioni cardinali pure (N, S, E, W) rispetto alle direzioni diagonali. Ad esempio, una direzione che è realmente NNE potrebbe essere percepita e ricordata come semplicemente N.

Un altro aspetto cognitivo rilevante è l'uso di landmark e sistemi di riferimento gerarchici. Gli esseri umani organizzano la conoscenza spaziale attorno a oggetti salienti (landmark) e utilizzano relazioni direzionali qualitative tra questi landmark piuttosto che coordinate metriche assolute.

I modelli mentali dello spazio incorporano spesso semplificazioni e idealizzazioni. Ad esempio, le relazioni direzionali possono essere rappresentate come se gli oggetti fossero punti, ignorando la loro estensione spaziale. Questo porta a rappresentazioni più compatte ma potenzialmente imprecise.

Applicazioni in Sistemi di Informazione Geografica

Le logiche direzionali qualitative trovano estese applicazioni nei Sistemi di Informazione Geografica (GIS). Le query spaziali espresse in linguaggio naturale spesso utilizzano termini direzionali qualitativi: "trova tutte le città a nord del fiume Po" o "mostra i ristoranti a est della stazione".

Un sistema GIS che supporta ragionamento direzionale qualitativo deve essere in grado di:

1. Estrarre relazioni direzionali da dati geometrici (punti, linee, poligoni)

2. Rappresentare efficientemente queste relazioni in un database spaziale
3. Rispondere a query che combinano vincoli direzionali con altri vincoli spaziali e attributi
4. Visualizzare i risultati in modo intuitivo per l'utente

L'estrazione di relazioni direzionali da geometrie può essere implementata attraverso algoritmi che calcolano la matrice di proiezione o determinano l'intersezione con coni direzionali. Per poligoni complessi, possono essere necessarie approssimazioni basate su bounding box, centroidi, o decomposizione in primitive più semplici.

L'indicizzazione di relazioni direzionali in database spaziali può sfruttare strutture come R-tree o quad-tree, estese per incorporare informazioni direzionali. Un approccio è quello di memorizzare per ogni oggetto la matrice di proiezione rispetto a un sistema di riferimento globale, permettendo calcoli efficienti di relazioni direzionali mediante operazioni matriciali.

Integrazione con Altre Relazioni Spaziali

Il ragionamento spaziale completo richiede l'integrazione di relazioni direzionali con altre relazioni qualitative, in particolare relazioni topologiche (come quelle del RCC-8) e relazioni di distanza.

Un esempio di integrazione è il sistema che combina direzione e topologia. Una relazione del tipo "la regione A è a nord di B e parzialmente sovrapposta ad essa" combina informazione direzionale (nord) e topologica (parzialmente sovrapposta). La consistenza di tali vincoli combinati richiede controlli che attraversano diverse dimensioni di conoscenza spaziale.

La rappresentazione formale di vincoli integrati può utilizzare tuple:

$\text{constraint}(A, B) = (\text{rel_topological}, \text{rel_directional}, \text{rel_distance})$

dove ogni componente specifica un aspetto della relazione spaziale. Ad esempio:

$\text{constraint}(A, B) = (\text{PO}, \{\text{N}, \text{NE}\}, \text{medium})$

indica che A e B si sovrappongono parzialmente, A è a nord o nord-est di B, e la distanza tra loro è media.

Il problema di soddisfacibilità per vincoli integrati è generalmente più complesso che per ciascuna dimensione singolarmente. Tecniche di decomposizione possono essere applicate: si verifica prima la consistenza per ciascuna dimensione separatamente, poi si controllano le interazioni tra dimensioni.

Reasoning Direzionale in Spazi Tridimensionali

L'estensione del Cardinal Direction Calculus a tre dimensioni introduce complessità significative. In uno spazio tridimensionale, le direzioni cardinali tradizionali devono essere integrate con direzioni verticali (su/giù o alto/basso).

Un sistema direzionale 3D completo potrebbe includere 27 regioni di base, derivate dalla partizione dello spazio mediante tre piani ortogonali. Le direzioni includerebbero:

- 8 direzioni ai vertici: NE-Alto, NE-Basso, NW-Alto, NW-Basso, SE-Alto, SE-Basso, SW-Alto, SW-Basso
- 12 direzioni sugli spigoli: N-Alto, N-Basso, E-Alto, E-Basso, S-Alto, S-Basso, W-Alto, W-Basso, N, E, S, W (nel piano orizzontale)
- 6 direzioni sulle facce: Alto, Basso, N (piano verticale), E (piano verticale), S (piano verticale), W (piano verticale)
- 1 direzione centrale: Stesso punto

La complessità combinatoria cresce rapidamente: per regioni estese, il numero di possibili combinazioni direzionali diventa proibitivo. Approcci pratici spesso decompongono il problema separando la componente orizzontale (direzioni cardinali) dalla componente verticale (alto/basso), ragionando su ciascuna indipendentemente e poi combinando i risultati.

Conclusioni e Prospettive

Le logiche direzionali qualitative forniscono un framework potente e cognitivamente plausibile per il ragionamento sulla direzione spaziale. La loro capacità di operare con informazione imprecisa e di scalare a problemi di grandi dimensioni le rende strumenti indispensabili per sistemi intelligenti che devono interagire con lo spazio.

Le direzioni future della ricerca includono l'integrazione più stretta con approcci di machine learning per apprendere relazioni direzionali da dati, lo sviluppo di algoritmi di ragionamento ancora più efficienti sfruttando parallelismo e strutture dati avanzate, e l'applicazione a nuovi domini come la robotica mobile, la realtà aumentata, e l'analisi di big data spaziali.

Logiche di Orientamento Intrinseco vs Estrinseco

Fondamenti Concettuali dei Sistemi di Riferimento Spaziali

La distinzione tra orientamento intrinseco ed estrinseco rappresenta uno dei problemi più profondi e cognitivamente rilevanti nel ragionamento spaziale qualitativo. Questa dicotomia riflette modi fondamentalmente diversi in cui gli agenti cognitivi - umani e artificiali - concettualizzano e comunicano informazioni sulla posizione e l'orientamento degli oggetti nello spazio.

Un sistema di riferimento spaziale è un framework concettuale e formale che permette di specificare la posizione e l'orientamento di entità spaziali. La scelta del sistema di riferimento

non è neutra: essa riflette assunzioni implicite sulla natura dello spazio, sui meccanismi cognitivi disponibili, e sugli scopi comunicativi o computazionali del sistema di ragionamento.

La letteratura sulla cognizione spaziale, iniziata con i lavori seminali di John O'Keefe sulla rappresentazione spaziale nell'ippocampo e proseguita con gli studi linguistici di Stephen Levinson, ha identificato tre principali tipi di sistemi di riferimento: intrinseco (o oggetto-centrato), relativo (o osservatore-centrato), e assoluto (o ambiente-centrato). Ciascuno di questi sistemi ha proprietà formali distintive, implicazioni cognitive specifiche, e domini di applicazione preferenziali.

Il Sistema di Riferimento Intrinseco: Formalizzazione Matematica

Il sistema di riferimento intrinseco, anche detto oggetto-centrato o allocentrico in alcune tradizioni, basa la descrizione spaziale sulle proprietà geometriche e funzionali intrinseche degli oggetti di riferimento. In questo sistema, espressioni come "davanti alla casa" si riferiscono al fronte architettonico della casa, indipendentemente dalla posizione o dall'orientamento di un osservatore.

Formalmente, un sistema di riferimento intrinseco per un oggetto A è specificato da una tupla $I_A = (o_A, \{v_1^A, v_2^A, v_3^A\}, \phi_A)$ dove:

- $o_A \in \mathbb{R}^3$ è l'origine del sistema di riferimento, tipicamente il centroide o un punto caratteristico di A
- $\{v_1^A, v_2^A, v_3^A\}$ è una base ortonormale di vettori che definiscono gli assi del sistema di coordinate locale
- $\phi_A: \text{Labels} \rightarrow \{v_1^A, v_2^A, v_3^A\}$ è una funzione che assegna etichette semantiche (come "fronte", "retro", "sinistra", "destra", "alto", "basso") agli assi

La determinazione del sistema intrinseco dipende dalle proprietà dell'oggetto. Per oggetti con funzionalità direzionale chiara (veicoli, edifici con facciata principale, animali), l'asse frontale v_1^A è allineato con la direzione funzionale primaria. Per oggetti simmetrici o senza orientamento intrinseco (sfere perfette, alcuni artefatti), il sistema intrinseco potrebbe non essere definito univocamente, richiedendo convenzioni esterne.

La trasformazione da coordinate globali (x, y, z) a coordinate intrinseche (x', y', z') per l'oggetto A è data dalla matrice di rotazione R_A e dalla traslazione t_A :

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R_A \begin{bmatrix} x \\ y \\ z \end{bmatrix} + t_A$$

dove R_A è una matrice 3×3 ortogonale che codifica l'orientamento del sistema intrinseco rispetto al sistema globale, e $t_A = -R_A \cdot o_A$ è il vettore di traslazione che porta l'origine globale nell'origine locale.

Il Sistema di Riferimento Relativo: Formalizzazione e Proprietà

Il sistema di riferimento relativo, o osservatore-centrato o egocentrico, basa le descrizioni spaziali sulla posizione e orientamento di un osservatore. In questo sistema, "la palla è a sinistra del tavolo" significa che la palla appare a sinistra del tavolo dal punto di vista dell'osservatore, e questa relazione cambierebbe se l'osservatore si muovesse.

Formalmente, un sistema di riferimento relativo è specificato da una tupla $R_{obs} = (o_{obs}, \theta_{obs}, \varphi_{obs})$ dove:

- $o_{obs} \in \mathbb{R}^3$ è la posizione dell'osservatore
- $\theta_{obs} \in [0, 2\pi)$ è l'angolo di orientamento orizzontale (azimuth) dell'osservatore
- $\varphi_{obs} \in [-\pi/2, \pi/2]$ è l'angolo di elevazione (pitch) dello sguardo dell'osservatore

Date due entità A e B, la relazione "B è a sinistra di A" nel sistema relativo è vera se, dalla prospettiva dell'osservatore che guarda verso A, B si trova nella regione sinistra del campo visivo.

Formalmente, consideriamo il vettore $v_{AB} = B - A$ che connette A a B. Questo vettore viene proiettato nel piano perpendicolare alla direzione di sguardo dell'osservatore. Sia $d_{obs} = (\cos(\theta_{obs})\cos(\varphi_{obs}), \sin(\theta_{obs})\cos(\varphi_{obs}), \sin(\varphi_{obs}))$ il vettore di direzione dell'osservatore. Il vettore proiettato è:

$$v_{proj} = v_{AB} - (v_{AB} \cdot d_{obs}) d_{obs}$$

Calcoliamo quindi il vettore "sinistra" dell'osservatore come:

$$v_{left} = d_{obs} \times (0, 0, 1) \text{ (normalizzato)}$$

B è "a sinistra" di A se $v_{proj} \cdot v_{left} > 0$.

Una proprietà cruciale del sistema relativo è la sua dipendenza dalla posizione dell'osservatore: se l'osservatore si muove o ruota, le relazioni spaziali relative cambiano anche se gli oggetti rimangono statici. Questo contrasta marcatamente con i sistemi intrinseco e assoluto.

Il Sistema di Riferimento Assoluto: Coordinate Geocentriche

Il sistema di riferimento assoluto utilizza coordinate fisse e globali, indipendenti sia dall'osservatore che dagli oggetti descritti. Il caso paradigmatico è l'uso di direzioni cardinali (nord, sud, est, ovest) basate sul campo magnetico terrestre o su convenzioni astronomiche.

Formalmente, un sistema assoluto è specificato da un sistema di coordinate globale fisso $A_{global} = (O, \{e_1, e_2, e_3\})$ dove O è un'origine convenzionale (ad esempio, un punto sulla superficie terrestre) e $\{e_1, e_2, e_3\}$ è una base ortonormale fissa. Tipicamente, e_1 punta a nord, e_2 punta a est, e e_3 punta verticalmente verso l'alto (o radialmente dalla superficie terrestre in sistemi geodetici precisi).

In un sistema assoluto, la relazione "B è a nord di A" è definita puramente dalla comparazione delle coordinate: se $y_B > y_A$ (dove y è la coordinata nord), allora B è a nord di A, indipendentemente da qualsiasi osservatore o dalle proprietà intrinseche di A e B.

Una complicazione pratica emerge quando si considerano distanze globali sulla superficie sferica della Terra. In questo caso, "nord" e "sud" sono definiti rispetto ai poli geografici, ma "est" e "ovest" sono definiti lungo cerchi di latitudine, che non sono geodetiche (cammini di distanza minima sulla sfera). Per piccole regioni, l'approssimazione piana è adeguata, ma per distanze globali è necessaria la geometria sferica o ellissoidale.

Trasformazioni tra Sistemi di Riferimento: Teoria Generale

Un problema centrale nelle logiche di orientamento è la trasformazione tra diversi sistemi di riferimento. Data una relazione spaziale espressa in un sistema, come possiamo convertirla in un altro sistema?

La trasformazione più complessa è quella tra sistema intrinseco e sistema relativo, poiché richiede la conoscenza di molteplici parametri. Supponiamo di avere:

- La relazione intrinseca $R_{\text{intr}}(A, B)$ = "B è davanti ad A" nel sistema intrinseco di A
- La posizione e orientamento di A nel sistema globale: (o_A, θ_A)
- La posizione e orientamento dell'osservatore: $(o_{\text{obs}}, \theta_{\text{obs}})$

Per determinare la relazione relativa $R_{\text{rel}}(A, B)$ dal punto di vista dell'osservatore, dobbiamo:

1. Convertire la relazione intrinseca in coordinate globali: Se B è "davanti" ad A nel sistema intrinseco, allora nel sistema globale B si trova nella direzione θ_A (l'orientamento frontale di A)
2. Calcolare l'angolo dal quale l'osservatore vede A: $\alpha_{\text{obs}_A} = \text{atan2}(y_A - y_{\text{obs}}, x_A - x_{\text{obs}})$
3. Calcolare l'angolo relativo tra la direzione osservatore→A e osservatore→B:
 $\alpha_{\text{obs}_B} = \text{atan2}(y_B - y_{\text{obs}}, x_B - x_{\text{obs}})$ $\Delta\alpha = \alpha_{\text{obs}_B} - \alpha_{\text{obs}_A}$
4. Determinare la relazione relativa basandosi su $\Delta\alpha$:
 - Se $\Delta\alpha \approx 0$: B è allineato con A dalla prospettiva dell'osservatore
 - Se $0 < \Delta\alpha < \pi$: B è a destra di A
 - Se $-\pi < \Delta\alpha < 0$: B è a sinistra di A

Questa trasformazione è non-monotona: piccole variazioni nelle posizioni possono portare a cambiamenti qualitativi nelle relazioni relative (ad esempio, da "a sinistra" a "a destra" attraverso una linea di allineamento).

Ambiguità e Risoluzione Contestuale

Un aspetto problematico dei sistemi di riferimento è l'ambiguità che emerge quando il contesto non specifica chiaramente quale sistema utilizzare. La frase "la palla è a sinistra del tavolo" potrebbe essere interpretata in almeno tre modi:

1. **Interpretazione Intrinseca:** La palla è a sinistra secondo il frame intrinseco del tavolo (il lato sinistro "canonico" del tavolo, se esiste)
2. **Interpretazione Relativa:** La palla è a sinistra dal punto di vista di chi parla o di un osservatore implicito
3. **Interpretazione Assoluta:** La palla è a ovest del tavolo (se "sinistra" è convenzionalmente associata a ovest in quel contesto)

La risoluzione di questa ambiguità richiede meccanismi di inferenza contestuale. La teoria della pertinenza (relevance theory) suggerisce che gli ascoltatori scelgono l'interpretazione che massimizza l'effetto cognitivo minimizzando lo sforzo di elaborazione.

Studi psicolinguistici hanno identificato vari fattori che influenzano la scelta del sistema di riferimento:

- **Proprietà dell'oggetto di riferimento:** Oggetti con un fronte chiaro (automobili, televisori) favoriscono l'interpretazione intrinseca
- **Salienza dell'osservatore:** Quando l'osservatore è menzionato esplicitamente o è cognitivamente saliente, l'interpretazione relativa è preferita
- **Contesto culturale:** Alcune culture (ad esempio, le comunità Guugu Yimithirr studiate da Levinson) usano prevalentemente sistemi assoluti anche in contesti dove altre culture userebbero sistemi relativi

Formalizzazione Logica Multi-Sistema

Per catturare formalmente la compresenza di molteplici sistemi di riferimento, possiamo estendere le logiche spaziali con operatori indicizzati per ciascun sistema. Il linguaggio della logica multi-sistema include:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \psi \mid [R_intr(A)]\varphi \mid [R_rel(obs)]\varphi \mid [R_abs]\varphi$$

dove:

- $[R_intr(A)]\varphi$ significa " φ è vero nel sistema intrinseco dell'oggetto A"
- $[R_rel(obs)]\varphi$ significa " φ è vero nel sistema relativo dell'osservatore obs"
- $[R_abs]\varphi$ significa " φ è vero nel sistema assoluto"

La semantica di questi operatori è definita su modelli estesi $M = (W, S_intr, S_rel, S_abs, V)$ dove:

- W è un insieme di configurazioni spaziali (mondi possibili)
- $S_{intr}: Objects \rightarrow (W \rightarrow W)$ assegna a ogni oggetto una funzione di trasformazione al suo sistema intrinseco
- $S_{rel}: Observers \rightarrow (W \rightarrow W)$ assegna a ogni osservatore una funzione di trasformazione al suo sistema relativo
- $S_{abs}: W \rightarrow W$ è la trasformazione al sistema assoluto
- V è la funzione di valutazione

Le condizioni di verità sono:

$M, w \models [R_{intr}(A)]\varphi$ se e solo se $M, S_{intr}(A)(w) \models \varphi$ $M, w \models [R_{rel}(obs)]\varphi$ se e solo se $M, S_{rel}(obs)(w) \models \varphi$ $M, w \models [R_{abs}]\varphi$ se e solo se $M, S_{abs}(w) \models \varphi$

Questo framework permette di esprimere e ragionare su relazioni tra sistemi. Ad esempio, possiamo esprimere che una relazione intrinseca implica una certa relazione assoluta date certe condizioni:

$[R_{intr}(A)]Front(B) \wedge Orient(A, N) \rightarrow [R_{abs}]North(B, A)$

che afferma: se B è davanti ad A nel sistema intrinseco di A , e A è orientato verso nord, allora B è a nord di A nel sistema assoluto.

Composizione e Consistenza tra Sistemi

La coesistenza di molteplici sistemi di riferimento nella stessa rappresentazione spaziale solleva questioni di consistenza. Un insieme di vincoli espressi in diversi sistemi potrebbe essere inconsistente se le trasformazioni tra sistemi non sono gestite correttamente.

Consideriamo il seguente scenario:

1. Nel sistema intrinseco di A : B è davanti ad A
2. Nel sistema relativo dell'osservatore O : B è a sinistra di A
3. Nel sistema assoluto: A è orientato a nord

Questi vincoli sono consistenti solo se la posizione dell'osservatore O è tale che, guardando verso A (che è orientato a nord), B appare a sinistra. Questo richiede che O sia posizionato in modo tale che l'angolo tra la direzione $O \rightarrow A$ e $O \rightarrow B$ sia negativo (indicando sinistra).

Il problema della consistenza multi-sistema è NP-completo in generale, poiché richiede di trovare un'assegnazione di posizioni e orientamenti che soddisfi tutti i vincoli attraverso le trasformazioni geometriche necessarie.

Algoritmi di consistenza locale possono essere applicati iterativamente:

1. Per ogni coppia di vincoli in sistemi diversi, calcolare le implicazioni nel sistema globale
2. Propagare queste implicazioni a tutti gli altri vincoli
3. Rilevare inconsistenze quando emergono contraddizioni

Variazione Linguistica e Universali Cognitivi

Gli studi cross-linguistici hanno rivelato notevoli variazioni nell'uso dei sistemi di riferimento spaziali. Stephen Levinson e collaboratori hanno documentato che:

- Lingue europee tipicamente usano tutti e tre i sistemi, con preferenza per il relativo in contesti di "tavolo" (piccola scala)
- Lingue come Tzeltal (Maya) usano predominantemente il sistema assoluto basato su caratteristiche geografiche (pendii montani)
- Lingue come Guugu Yimithirr (Australia) usano esclusivamente sistemi assoluti anche per oggetti di piccola scala

Tuttavia, esistono anche universali cognitivi. Tutti gli esseri umani sembrano avere la capacità di utilizzare tutti e tre i tipi di sistemi, anche se le norme linguistiche e culturali favoriscono alcuni a scapito di altri. Questa plasticità cognitiva suggerisce che i sistemi di riferimento non sono moduli innati rigidi ma strategie flessibili che possono essere apprese e modificate.

Dal punto di vista formale, questa variazione può essere modellata come una gerarchia di preferenze sui sistemi:

Pref_culture: {intrinsic, relative, absolute} $\rightarrow$ [0, 1]

dove ogni cultura assegna pesi diversi ai sistemi, influenzando quale sistema viene scelto in situazioni ambigue.

Rappresentazioni Ibride e Multi-Risoluzione

Sistemi pratici di ragionamento spaziale spesso impiegano rappresentazioni ibride che combinano molteplici sistemi di riferimento a diverse risoluzioni e per diverse porzioni dello spazio.

Ad esempio, un sistema di navigazione robotica potrebbe usare:

- Sistema assoluto per navigazione globale (basato su GPS e mappe)
- Sistema relativo per evitamento ostacoli immediate (sensori centrati sul robot)
- Sistema intrinseco per manipolazione di oggetti (coordinate relative agli oggetti manipolati)

La transizione tra questi sistemi deve essere gestita esplicitamente. Un framework gerarchico potrebbe strutturare lo spazio in livelli:

Livello Globale (sistema assoluto):

- Coordinate geografiche
- Mappe topologiche di larga scala

Livello Locale (sistema relativo):

- Coordinate robot-centriche
- Sensori propriocettivi ed esteroceettivi

Livello Oggetto (sistema intrinseco):

- Coordinate oggetto-centriche
- Proprietà funzionali e affordances

Ogni livello ha un proprio vocabolario di relazioni spaziali e operazioni permesse. Le trasformazioni tra livelli sono esplicitate mediante funzioni di lift e lower che convertono rappresentazioni tra granularità diverse.

Apprendimento Automatico di Sistemi di Riferimento

Un'area emergente di ricerca riguarda l'apprendimento automatico di sistemi di riferimento da dati. Invece di definire manualmente quali oggetti hanno quale sistema intrinseco, algoritmi di machine learning possono inferire sistemi di riferimento appropriati da osservazioni di linguaggio spaziale in contesto.

Un approccio utilizza reti neurali per apprendere embeddings di oggetti in uno spazio latente dove la geometria riflette i sistemi di riferimento impliciti. Ad esempio, oggetti con forti proprietà intrinseche (come automobili) avranno embeddings che codificano direzionalità, mentre oggetti senza orientamento intrinseco (come pietre sferiche) avranno embeddings isotropi.

Formalmente, un modello neurale $f_\theta: \text{Objects} \times \text{Contexts} \rightarrow \text{RefSystems}$ apprende una funzione parametrizzata da θ che mappa oggetti e contesti a distribuzioni di probabilità su sistemi di riferimento:

$$P(R = \text{intrinsic} \mid \text{obj}, \text{context}) = \text{softmax}(f_\theta(\text{obj}, \text{context}))[0] \\ P(R = \text{relative} \mid \text{obj}, \text{context}) = \text{softmax}(f_\theta(\text{obj}, \text{context}))[1] \\ P(R = \text{absolute} \mid \text{obj}, \text{context}) = \text{softmax}(f_\theta(\text{obj}, \text{context}))[2]$$

L'addestramento di tale modello su corpora annotati di descrizioni spaziali permette di catturare le preferenze contestuali e le proprietà oggetto-specifiche che determinano la scelta del sistema di riferimento.

Applicazioni in Robotica e Interazione Uomo-Robot

La distinzione tra sistemi di riferimento è cruciale per l'interazione naturale uomo-robot. Quando un umano dice a un robot "prendi l'oggetto a sinistra", il robot deve disambiguare se "sinistra" si riferisce:

1. Alla sinistra del robot stesso (sistema relativo robot-centric)
2. Alla sinistra dell'umano parlante (sistema relativo umano-centric)
3. A una sinistra convenzionale della scena o tavolo (sistema intrinseco della scena)

Le strategie di risoluzione includono:

Euristica basata su salienza: Se il robot è appena stato menzionato o è l'agente dell'azione, usare il sistema robot-centrico. Se l'umano è stato appena menzionato, usare il sistema umano-centrico.

Sguardo e gesti: Tracciare la direzione dello sguardo o dei gesti indicativi dell'umano per inferire il sistema di riferimento inteso.

Dialogo chiarificatore: In caso di ambiguità residua, il robot può chiedere esplicitamente "intendi alla mia sinistra o alla tua sinistra?"

Formalmente, un robot ragiona su una distribuzione di probabilità su sistemi di riferimento, aggiornata dinamicamente basandosi su evidenza contestuale:

$$P(R \mid \text{utterance}, \text{context}) \propto P(\text{utterance} \mid R, \text{context}) P(R \mid \text{context})$$

dove $P(\text{utterance} \mid R, \text{context})$ è la likelihood che l'enunciato sia stato prodotto assumendo il sistema R, e $P(R \mid \text{context})$ è la probabilità a priori del sistema R dato il contesto.

Conclusioni e Direzioni Future

La distinzione tra sistemi di riferimento intrinseci, relativi e assoluti è fondamentale per comprendere e formalizzare il ragionamento spaziale. Ogni sistema ha proprietà formali distintive, vantaggi computazionali specifici, e basi cognitive proprie.

Le sfide future includono lo sviluppo di teorie unificate che integrino seamlessly i tre sistemi, algoritmi efficienti per trasformazioni e consistenza multi-sistema, e modelli computazionali che catturano la flessibilità cognitiva umana nel passare tra sistemi.

Applicazioni promettenti spaziano dalla robotica sociale ai sistemi di realtà aumentata, dai GIS intelligenti agli assistenti virtuali che comprendono descrizioni spaziali naturali in tutta la loro ricchezza e ambiguità.

Qualitative Spatial Reasoning (QSR)

Fondamenti Epistemologici del Ragionamento Spaziale Qualitativo

Il Qualitative Spatial Reasoning (QSR) rappresenta un paradigma

fondamentale nell'intelligenza artificiale e nella rappresentazione della conoscenza che si distingue dalle rappresentazioni metriche quantitative per la sua enfasi su relazioni categoriali, discrete e simboliche tra entità spaziali. L'assunto centrale del QSR è che molti compiti di ragionamento spaziale non richiedono informazioni metriche precise, e che rappresentazioni qualitative possono essere più robuste, comprensibili, e computazionalmente trattabili.

La motivazione epistemologica del QSR deriva da diverse osservazioni convergenti. Prima, la cognizione spaziale umana opera frequentemente a livello qualitativo: descriviamo oggetti come "vicini" o "lontani", "dentro" o "fuori", "davanti" o "dietro", senza necessariamente quantificare distanze o angoli precisi. Secondo, l'informazione spaziale disponibile in molti contesti pratici è intrinsecamente qualitativa o afflitta da incertezze che rendono le rappresentazioni metriche inappropriate. Terzo, il ragionamento qualitativo può essere più efficiente computazionalmente per certi tipi di inferenze dove dettagli metrici sono irrilevanti.

Il QSR si fonda su alcuni principi metodologici chiave: la discretizzazione dello spazio continuo in un insieme finito di relazioni qualitative distintive; la definizione di operazioni algebriche (composizione, inversione) su queste relazioni; lo studio di proprietà formali come consistenza e completezza; e lo sviluppo di algoritmi per problemi di ragionamento come soddisfacibilità, inferenza, e planning.

Il Modello 9-Intersection di Egenhofer e Herring

Il 9-Intersection Model, sviluppato da Max Egenhofer e colleghi all'inizio degli anni Novanta, rappresenta uno dei formalismi più raffinati per caratterizzare relazioni topologiche tra regioni spaziali. Questo modello estende e raffina il Region Connection Calculus (RCC-8) fornendo una base matematicamente rigorosa basata sulla topologia point-set.

Il modello si fonda sulla decomposizione topologica di una regione in tre componenti mutuamente disgiunte: l'interno (A°), il confine (∂A), e l'esterno (A^-). Per una regione $A \subset \mathbb{R}^n$:

- **Interno:** $A^\circ = \{x \in A : \text{esiste } \varepsilon > 0 \text{ tale che } B_\varepsilon(x) \subseteq A\}$, dove $B_\varepsilon(x)$ denota la palla aperta di raggio ε centrata in x
- **Confine:** $\partial A = \text{cl}(A) \cap \text{cl}(\mathbb{R}^n \setminus A)$, dove cl denota la chiusura topologica
- **Esterno:** $A^- = (\text{cl}(A))^c$, il complemento della chiusura di A

La 9-intersection matrix tra due regioni A e B è una matrice 3×3 i cui elementi indicano l'intersezione tra ciascuna componente di A e ciascuna componente di B :

	B°	∂B	B^-
A°	$A^\circ \cap B^\circ$	$A^\circ \cap \partial B$	$A^\circ \cap B^-$
∂A	$\partial A \cap B^\circ$	$\partial A \cap \partial B$	$\partial A \cap B^-$
A^-	$A^- \cap B^\circ$	$A^- \cap \partial B$	$A^- \cap B^-$

Ogni cella contiene un valore booleano: $\emptyset$ (vuoto) o $\neg\emptyset$ (non vuoto), indicando se l'intersezione corrispondente è vuota o meno. Teoricamente, questo genera $2^9 = 512$ possibili matrici, ma vincoli topologici riducono drasticamente il numero di configurazioni realizzabili.

Per regioni semplici (regioni regolarmente chiuse, cioè regioni che coincidono con la chiusura del loro interno) in $\mathbb{R}^2$, solo 8 delle 512 configurazioni sono realizzabili. Queste otto relazioni corrispondono esattamente alle relazioni del RCC-8:

Disjoint (DC): A e B sono completamente separati

$\emptyset \quad \emptyset \quad \neg \emptyset$
 $\emptyset \quad \emptyset \quad \neg \emptyset$
 $\neg \emptyset \quad \neg \emptyset \quad \neg \emptyset$

1.

Externally Connected (EC): I confini si toccano ma gli interni sono disgiunti

$\emptyset \quad \neg \emptyset \quad \neg \emptyset$
 $\neg \emptyset \quad \emptyset \quad \neg \emptyset$
 $\neg \emptyset \quad \neg \emptyset \quad \neg \emptyset$

2.

Partially Overlapping (PO): Gli interni si intersecano ma nessuna regione contiene l'altra

$\neg \emptyset \quad \neg \emptyset \quad \neg \emptyset$
 $\neg \emptyset \quad \emptyset \quad \neg \emptyset$
 $\neg \emptyset \quad \neg \emptyset \quad \neg \emptyset$

3.

Equal (EQ): Le due regioni coincidono

$\neg \emptyset \quad \neg \emptyset \quad \emptyset$
 $\neg \emptyset \quad \emptyset \quad \emptyset$
 $\emptyset \quad \emptyset \quad \neg \emptyset$

4.

Tangential Proper Part (TPP): A è contenuto in B e i confini si toccano

$\neg \emptyset \quad \neg \emptyset \quad \emptyset$
 $\neg \emptyset \quad \emptyset \quad \emptyset$
 $\neg \emptyset \quad \neg \emptyset \quad \neg \emptyset$

5.

Non-Tangential Proper Part (NTPP): A è strettamente contenuto in B senza toccare il confine

$\neg \emptyset \quad \emptyset \quad \emptyset$
 $\emptyset \quad \emptyset \quad \emptyset$
 $\neg \emptyset \quad \neg \emptyset \quad \neg \emptyset$

6.

Tangential Proper Part Inverse (TPP⁻¹): B è contenuto in A e i confini si toccano

 $\neg \emptyset \neg \emptyset \neg \emptyset$
 $\neg \emptyset \emptyset \neg \emptyset$
 $\emptyset \emptyset \neg \emptyset$

7.

Non-Tangential Proper Part Inverse (NTPP⁻¹): B è strettamente contenuto in A

 $\neg \emptyset \neg \emptyset \neg \emptyset$
 $\emptyset \emptyset \neg \emptyset$
 $\emptyset \emptyset \neg \emptyset$

8.

L'estensione di questo modello a regioni con "buchi" (regioni non semplici) aumenta il numero di relazioni distinguibili. Per regioni con un buco, il numero di relazioni topologicamente distinte sale a 23. Per regioni con confini multipli-connessi, il numero cresce ulteriormente.

II Double Cross Calculus: Relazioni tra Linee

Il Double Cross Calculus, sviluppato da Freksa e collaboratori, fornisce un formalismo qualitativo per rappresentare relazioni tra linee orientate o segmenti direzionali nel piano. Questo calcolo è particolarmente rilevante per applicazioni come la navigazione, dove percorsi e traiettorie sono naturalmente rappresentati come linee orientate.

Una linea orientata l in R^2 può essere caratterizzata da un punto di inizio $s(l)$, un punto finale $e(l)$, e implicitamente dalla direzione da $s(l)$ a $e(l)$. Il Double Cross Calculus considera due primitive geometriche fondamentali per ogni coppia di linee orientate l_1 e l_2 :

1. La posizione del punto finale $e(l_1)$ rispetto alla linea l_2
2. La posizione del punto iniziale $s(l_2)$ rispetto alla linea l_1

Per determinare la posizione di un punto p rispetto a una linea orientata l , il calcolo utilizza il concetto di "cross product" o prodotto vettoriale. Dato il vettore direzione $v_l = e(l) - s(l)$ e il vettore da $s(l)$ a p : $v_p = p - s(l)$, il prodotto vettoriale (in 2D, equivalente alla componente z del prodotto vettoriale 3D) è:

$$\text{cross}(v_l, v_p) = (e(l).x - s(l).x)(p.y - s(l).y) - (e(l).y - s(l).y)(p.x - s(l).x)$$

Il segno di questo valore determina la posizione relativa:

- $\text{cross} > 0$: p è a sinistra della linea l (nella direzione di orientamento)
- $\text{cross} = 0$: p è sulla linea l
- $\text{cross} < 0$: p è a destra della linea l

Utilizzando questa primitiva due volte (per $e(l_1)$ rispetto a l_2 e per $s(l_2)$ rispetto a l_1), il Double Cross Calculus distingue $5 \times 5 = 25$ relazioni di base, dove ogni dimensione ha 5 possibilità:

- left (l): Il punto è strettamente a sinistra
- right (r): Il punto è strettamente a destra
- start (s): Il punto coincide con il punto iniziale della linea
- end (e): Il punto coincide con il punto finale della linea
- between (b): Il punto è sulla linea tra inizio e fine

Tuttavia, non tutte le 25 combinazioni sono geometricamente realizzabili. Un'analisi combinatoria rivela che solo 13 configurazioni possono effettivamente occorrere per segmenti generici nel piano. Queste 13 relazioni includono:

1. **Separate sinistra-sinistra** (ll): Entrambi i punti chiave sono a sinistra
2. **Croce sinistra-destra** (lr): $e(l_1)$ è a sinistra di l_2 , $s(l_2)$ è a destra di l_1 (le linee si incrociano)
3. **Parallele stesso-verso** (bb): Le linee sono parallele e orientate nello stesso verso
4. **Parallele verso-opposto** (bb'): Le linee sono parallele ma orientate in versi opposti
5. **T-junction inizio-inizio** (ss): I punti iniziali coincidono
6. **T-junction fine-fine** (ee): I punti finali coincidono
7. **Collineari orientate-insieme** (b-overlap): Le linee sono sulla stessa retta, con sovrapposizione, orientate ugualmente

E altre configurazioni che catturano casi degeneri e speciali.

La tavola di composizione per il Double Cross Calculus specifica come due relazioni si compongono per formare una terza. Ad esempio, se l_1 incrocia l_2 da sinistra a destra (lr), e l_2 incrocia l_3 da sinistra a destra (lr), allora la relazione tra l_1 e l_3 è indeterminata senza ulteriori informazioni (potrebbe essere qualsiasi delle relazioni, a seconda delle posizioni esatte).

La composizione è formalmente definita come:

$$R(l_1, l_3) \supseteq R(l_1, l_2) \circ R(l_2, l_3)$$

dove il lato destro fornisce un limite superiore (un sovrainsieme) della relazione effettiva. La path-consistency in un network di vincoli basato sul Double Cross Calculus può essere imposta applicando ripetutamente questa regola di composizione.

Il Dipole Calculus: Oggetti Orientati

Il Dipole Calculus, sviluppato da Schlieder, estende il ragionamento qualitativo a oggetti che hanno sia posizione che orientamento intrinseco. Un dipolo è rappresentato come un segmento orientato definito da due punti distinti: un punto iniziale (tail) e un punto finale (head). Questo è particolarmente utile per modellare entità come veicoli, frecce, o qualsiasi oggetto con una direzione naturale.

Formalmente, un dipolo d è una coppia ordinata $d = (\text{tail}(d), \text{head}(d))$ dove $\text{tail}(d), \text{head}(d) \in \mathbb{R}^2$ e $\text{tail}(d) \neq \text{head}(d)$. La lunghezza e la direzione del dipolo sono implicite nella rappresentazione.

Il Dipole Calculus definisce relazioni qualitative tra dipoli basandosi sulla posizione relativa dei quattro punti coinvolti (due per ciascun dipolo). Per due dipoli d_1 e d_2 , il calcolo considera:

1. La posizione di $\text{tail}(d_2)$ rispetto a d_1 (considerato come linea orientata)
2. La posizione di $\text{head}(d_2)$ rispetto a d_1
3. Analogamente per le posizioni di $\text{tail}(d_1)$ e $\text{head}(d_1)$ rispetto a d_2

Utilizzando le stesse primitive del Double Cross Calculus (left, right, on, start, end), ma applicandole a entrambe le estremità di ciascun dipolo, il Dipole Calculus genera un insieme più ricco di relazioni.

Una rappresentazione compatta utilizza matrici 2×2 :

	d_1	
	tail	head
d_2	tail	A B
	head	C D

dove $A, B, C, D \in \{l, r, s, e, b\}$ indicano le posizioni relative. Tuttavia, vincoli geometrici riducono drasticamente le combinazioni realizzabili.

Il sistema completo del Dipole Calculus distingue 72 relazioni di base, che possono essere raggruppate in classi di equivalenza basate su simmetrie e trasformazioni geometriche. Queste relazioni catturano configurazioni come:

- **Paralleli con stesso verso:** I dipoli sono paralleli e puntano nella stessa direzione generale
- **Paralleli con verso opposto:** Paralleli ma orientati in direzioni opposte
- **Convergenti:** I dipoli, se estesi, si incontrerebbero davanti
- **Divergenti:** I dipoli puntano in direzioni che si allontanano
- **Ortogonal:** I dipoli formano approssimativamente un angolo retto
- **Sequenziali:** Un dipolo segue l'altro in una configurazione simile a una catena

La composizione di relazioni nel Dipole Calculus è computazionalmente più complessa che nei calcoli più semplici. La tavola di composizione è una struttura 72×72 , e molte composizioni risultano in disgiunzioni di molteplici relazioni possibili. L'imposizione di path-consistency può ridurre l'incertezza ma non garantisce la consistenza globale.

Un'applicazione importante del Dipole Calculus è nel ragionamento su traiettorie e movimento. Se un oggetto in movimento può essere rappresentato come un dipolo (posizione attuale $\rightarrow$ posizione prevista), allora il Dipole Calculus permette di ragionare su configurazioni di multipli oggetti in movimento, prevedere collisioni, o pianificare percorsi che evitano interferenze.

Integrazione Multi-Aspetto: Topologia + Direzione + Distanza

Un ragionamento spaziale completo e realistico richiede l'integrazione di molteplici aspetti qualitativi: relazioni topologiche, direzionali, e di distanza. Ciascun aspetto cattura informazioni complementari che, combinate, forniscono una rappresentazione più ricca e discriminante.

Un framework integrato utilizza tuple di relazioni:

$$R(A, B) = (R_topo(A, B), R_dir(A, B), R_dist(A, B))$$

dove:

- $R_topo \in \text{RCC-8}$ (o 9-Intersection Model esteso)
- $R_dir \in \text{DIR}_9$ (o altro calcolo direzionale)
- $R_dist \in \{\text{touching, close, medium, far}\}$ (o altro calcolo di distanza qualitativa)

La consistenza di questi vincoli combinati è più restrittiva che per ciascun aspetto singolarmente. Ad esempio:

- Se $R_topo(A, B) = \text{DC}$ (disgiunte) e $R_dist(A, B) = \text{touching}$, c'è una contraddizione
- Se $R_topo(A, B) = \text{EC}$ (esternamente connesse) allora necessariamente $R_dist(A, B) = \text{touching}$
- Se $R_dir(A, B) = \{N, NE\}$ e $R_topo(A, B) = \text{NTPP}$ (A dentro B), ci sono vincoli sulla forma e posizione relativa

La propagazione di vincoli in un sistema multi-aspetto richiede regole di inferenza che attraversano le dimensioni. Regole tipiche includono:

Regola Topologia→Distanza:

- Se $\text{NTPP}(A, B)$ allora $\text{touching}(A, B)$ (confini a contatto implica vicinanza)
- Se $\text{DC}(A, B)$ e $\text{small}(A)$ e $\text{small}(B)$ allora $\text{close}(A, B) \vee \text{medium}(A, B) \vee \text{far}(A, B)$ (disgiunti non dice nulla sulla distanza da solo)

Regola Direzione+Distanza→Topologia:

- Se $N(A, B)$ e $\text{far}(A, B)$ allora $\text{DC}(A, B)$ (oggetti lontani in direzione cardinale sono disgiunti)
- Se $\{N, NE, E\}(A, B)$ e $\text{touching}(A, B)$ allora $\text{EC}(A, B) \vee \text{PO}(A, B)$ (vicinanza con direzione multipla suggerisce sovrapposizione)

Un problema fondamentale è il calcolo della tavola di composizione multi-aspetto. Data la composizione nelle singole dimensioni:

$$R_1 = (T_1, D_1, \text{Dist}_1) \text{ e } R_2 = (T_2, D_2, \text{Dist}_2)$$

La composizione $R_1 \circ R_2$ è:

$$R_1 \circ R_2 \subseteq (T_1 \circ T_2, D_1 \circ D_2, \text{Dist}_1 \circ \text{Dist}_2)$$

Ma questa è solo un'approssimazione superiore. Le interazioni tra dimensioni possono restringere ulteriormente le possibilità. Ad esempio, se $T_1 \circ T_2$ include PO (sovrapposizione) ma $D_1 \circ D_2$ include solo direzioni cardinali pure (N, S, E, W) senza sovrapposizioni direzionali, c'è una contraddizione che deve essere risolta eliminando combinazioni impossibili.

Algoritmi di Ragionamento: Path-Consistency e oltre

Gli algoritmi di ragionamento nel QSR si concentrano principalmente su problemi di consistenza e di inferenza. Il problema di consistenza chiede: dato un insieme di vincoli qualitativi, esiste una configurazione spaziale che li soddisfa tutti? Il problema di inferenza chiede: data una base di conoscenza di vincoli, quali nuovi vincoli possono essere derivati?

L'algoritmo fondamentale è la **Path-Consistency (PC)**, che impone una forma locale di consistenza. Un network di vincoli è path-consistent se per ogni tripla di variabili (i, j, k), il vincolo diretto tra i e j è consistente con il percorso indiretto attraverso k:

$$R_{ij} \subseteq R_{ij} \cap (R_{ik} \circ R_{kj})$$

L'algoritmo PC-2 implementa questo controllo iterativamente:

PC-2(V, C):

$Q = \{(i,j,k) : i,j,k \in V, i \neq j \neq k \neq i\}$

while $Q \neq \emptyset$:

 seleziona e rimuovi (i,j,k) da Q

$R_{ij_new} = R_{ij} \cap (R_{ik} \circ R_{kj})$

 if $R_{ij_new} = \emptyset$:

 return INCONSISTENTE

 if $R_{ij_new} \subset R_{ij}$:

$R_{ij} = R_{ij_new}$

 aggiungi a Q tutte le triple (i,j,m) e (m,j,i) per ogni $m \neq i,j$

return PATH-CONSISTENTE

La complessità temporale di PC-2 è $O(n^3 \cdot k^3)$ dove n è il numero di variabili e k è il numero massimo di relazioni di base in una disgiunzione. La complessità spaziale è $O(n^2)$.

Tuttavia, path-consistency non garantisce consistenza globale per tutti i calcoli qualitativi. Alcuni calcoli, come il Cardinal Direction Calculus con regioni estese, non sono path-consistent completi: esistono network path-consistent che non ammettono realizzazioni geometriche.

Per questi calcoli, algoritmi più sofisticati sono necessari:

Backtracking con Propagazione: Assegna valori alle variabili una alla volta, propagando vincoli dopo ogni assegnazione e facendo backtrack in caso di inconsistenza.

*A-based Consistency Checking**: Utilizza euristiche di ricerca informata per esplorare lo spazio delle assegnazioni possibili, guidato da stime della probabilità di soddisfacibilità.

SAT/CSP Encoding: Codifica il problema di consistenza come un problema SAT (soddisfacibilità booleana) o CSP generale e utilizza solver ottimizzati.

Complessità Computazionale del QSR

Lo studio della complessità computazionale del QSR ha rivelato una gamma di risultati, dipendenti dal calcolo specifico e dal problema considerato.

Per il **RCC-8**:

- Il problema di consistenza per network completi è NP-completo
- Path-consistency può essere verificata in tempo polinomiale $O(n^3)$
- Path-consistency è decisionale per una sottoclasse (la classe dei network atomici e alcuni altri)
- Il problema di soddisfacibilità minimale (trovare la realizzazione con numero minimo di regioni) è NP-hard

Per il **Cardinal Direction Calculus**:

- Il problema di consistenza con regioni estese è NP-completo
- Il problema diventa polinomiale se le regioni sono ristrette a essere rettangoli allineati agli assi
- L'imposizione di path-consistency è polinomiale ma non decisionale

Per l'**Interval Algebra** (Allen's temporal intervals, applicabile anche a intervalli spaziali unidimensionali):

- Il problema di consistenza è NP-completo in generale
- Esiste un sottoclasse "ORD-Horn" per cui la consistenza è decidibile in tempo polinomiale
- Il problema di consistenza per il frammento convesso è polinomiale

Per il **Double Cross Calculus**:

- Il problema di consistenza è NP-completo
- Non esiste sottoclasse maximal tractable conosciuta (a parte casi triviali)

Questi risultati guidano la scelta di algoritmi e rappresentazioni per applicazioni pratiche. Per problemi piccoli o quando è necessaria completezza, algoritmi esponenziali sono accettabili. Per problemi grandi, si utilizzano approssimazioni polinomiali (path-consistency) accettando incompletezza.

Estensioni Probabilistiche e Fuzzy

Il QSR deterministico assume che le relazioni spaziali siano note con certezza e siano nettamente categoriali. In molte applicazioni reali, tuttavia, esiste incertezza sulle relazioni o le relazioni stesse sono graduali.

QSR Probabilistico associa a ogni relazione una probabilità:

$P(R(A, B) = r)$ per ogni $r \in \text{Relations}$

Dato un insieme di osservazioni probabilistiche, il problema di inferenza chiede di calcolare la distribuzione di probabilità su configurazioni spaziali possibili o su relazioni derivate.

Un approccio utilizza reti bayesiane dove i nodi rappresentano relazioni spaziali e gli archi rappresentano dipendenze. Le tavole di probabilità condizionale catturano come la probabilità di una relazione dipende dalle probabilità di altre relazioni attraverso composizione.

Ad esempio: $P(R_{13} = r \mid R_{12} = r_1, R_{23} = r_2)$ modella la probabilità della composizione.

Algoritmi di inferenza come belief propagation o MCMC (Markov Chain Monte Carlo) possono essere utilizzati per calcolare distribuzioni marginali e condizionali.

QSR Fuzzy sostituisce relazioni nette con gradi di appartenenza:

$\mu_r(A, B) \in [0, 1]$ per ogni relazione r

Ad esempio, due regioni potrebbero essere "parzialmente sovrapposte" con grado 0.7 e "esternamente connesse" con grado 0.3, se la loro configurazione è al confine tra queste categorie.

Le operazioni di composizione e inversione sono estese usando logica fuzzy:

$\mu_{r_1 \circ r_2}(A, C) = \max_B \min(\mu_{r_1}(A, B), \mu_{r_2}(B, C))$

Questa formula usa operatori min/max (T-norma e T-conorma di Zadeh), ma altre scelte sono possibili (prodotto, Łukasiewicz).

La path-consistency fuzzy richiede che:

$\mu_{r_1 \square}(i, j) \geq \max_k \min(\mu_{r_1 \square}(i, k), \mu_{r_2 \square \square}(k, j))$

Algoritmi iterativi possono rafforzare i gradi di appartenenza fino a raggiungere un punto fisso che rappresenta la massima consistenza fuzzy.

Applicazioni e Case Studies

Il QSR trova applicazioni in numerosi domini:

Geographic Information Systems (GIS): Query spaziali qualitative su database geografici, analisi di relazioni tra feature geografiche, planning di percorsi basato su landmark e relazioni qualitative.

Robotica Mobile: Mapping qualitativo dove il robot costruisce una rappresentazione simbolica dello spazio esplorato, navigazione basata su landmark, ragionamento su azioni e loro effetti spaziali.

Computer Vision: Reasoning sulla disposizione di oggetti rilevati in scene visive, comprensione di scene descritte linguisticamente, integrazione di riconoscimento di oggetti con ragionamento spaziale.

Natural Language Processing: Interpretazione di descrizioni spaziali in testo, generazione di descrizioni spaziali naturali da rappresentazioni simboliche, question answering su informazioni spaziali.

Archeologia e Analisi Storica: Reasoning su disposizioni di artefatti, interpretazione di mappe e descrizioni storiche imprecise, ricostruzione di configurazioni spaziali da evidenze frammentarie.

Design e Architettura: Specifica di vincoli qualitativi su disposizioni di stanze e componenti, verifica di consistenza di design preliminari, generazione automatica di layout che soddisfano vincoli qualitativi.

Un case study esemplare è il sistema **SPATIAL ANALYZER** sviluppato per supportare archeologi nell'interpretazione di siti di scavo. Il sistema permette agli utenti di specificare vincoli qualitativi del tipo:

- La tomba A è a nord della struttura B
- L'artefatto C è dentro la tomba A
- La struttura D è vicina ma non connessa alla struttura B

Il sistema verifica la consistenza di questi vincoli, suggerisce ricostruzioni possibili, e identifica informazioni mancanti critiche che, se acquisite, risolverebbero ambiguità.

Direzioni Future e Sfide Aperte

Nonostante i progressi sostanziali, il QSR affronta ancora diverse sfide aperte:

Integrazione Multi-Scala: Come rappresentare e ragionare su relazioni spaziali che span molteplici scale, dalla nanoscala alla scala geografica, in modo uniforme?

QSR Dinamico: Come estendere i formalismi a situazioni dove le relazioni cambiano nel tempo? Come ragionare su processi spaziali e traiettorie in modo qualitativo?

Machine Learning per QSR: Come apprendere automaticamente relazioni qualitative e tavole di composizione da dati? Come integrare apprendimento profondo con reasoning simbolico qualitativo?

QSR in Spazi Non-Euclidei: Come estendere i formalismi a spazi curvi, reti, o spazi topologici più generali?

Spiegabilità: Come rendere trasparente e comprensibile il ragionamento qualitativo, specialmente quando integrato con sistemi di machine learning?

Queste domande definiscono l'agenda di ricerca per i prossimi decenni, con potenziali applicazioni in intelligenza artificiale generale, robotica avanzata, e sistemi di supporto decisionale complessi.

Logiche per il Reasoning su Forme

Fondamenti Matematici della Teoria delle Forme

Il ragionamento formale sulle forme geometriche rappresenta un'area fondamentale che interseca la geometria computazionale, la topologia, e la teoria della rappresentazione della conoscenza. Una forma può essere concepita come una classe di equivalenza di regioni spaziali sotto trasformazioni geometriche appropriate (traslazioni, rotazioni, scalature), catturando le proprietà geometriche intrinseche indipendenti da posizione, orientamento, e dimensione assoluta.

La formalizzazione matematica rigorosa delle forme richiede la definizione di uno spazio delle forme e di metriche o pseudo-metriche su tale spazio. Il **shape space** S per regioni bidimensionali in $\mathbb{R}^2$ può essere definito come l'insieme quoziente:

$$S = \{\text{regioni in } \mathbb{R}^2\} / \sim$$

dove $\sim$ è una relazione di equivalenza definita da trasformazioni geometriche. La scelta delle trasformazioni determina quale nozione di "stessa forma" viene catturata:

- **Similitudine:** Traslazione + rotazione + scaling uniforme (preserva angoli e rapporti di lunghezza)
- **Affinità:** Trasformazioni affini (preserva parallelismo e rapporti di divisione)
- **Proiettiva:** Trasformazioni proiettive (preserva cross-ratios)
- **Topologica:** Omeomorfismi (preserva solo proprietà topologiche come connessione e genere)

Per il ragionamento qualitativo, la nozione di similitudine è spesso più appropriata, catturando l'intuizione che due oggetti hanno "la stessa forma" se uno può essere ottenuto dall'altro tramite movimento rigido e ridimensionamento proporzionale.

Shape Algebras: Strutture Algebriche sulle Forme

Le shape algebras forniscono un framework formale per definire operazioni sulle forme e studiare le loro proprietà algebriche. Una shape algebra è una struttura $(S, +, \cdot, 0, 1, \leq)$ dove:

- S è l'insieme delle forme
 - $+$ è un'operazione binaria di "unione" o "combinazione" di forme

- $\cdot$ è un'operazione di "intersezione" o "sovrapposizione" di forme
- 0 è la forma vuota (elemento neutro per $+$)
- 1 è la forma universale (tutto lo spazio)
- $\leq$ è una relazione di ordine parziale interpretabile come "è contenuta in" o "è parte di"

Assiomi fondamentali includono:

Commutatività: $a + b = b + a$ e $a \cdot b = b \cdot a$

Associatività: $(a + b) + c = a + (b + c)$ e $(a \cdot b) \cdot c = a \cdot (b \cdot c)$

Distributività: $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$

Idempotenza: $a + a = a$ e $a \cdot a = a$

Assorbimento: $a + (a \cdot b) = a$ e $a \cdot (a + b) = a$

Complementazione: Per ogni a esiste a' tale che $a + a' = 1$ e $a \cdot a' = 0$

Queste proprietà definiscono una **Boolean algebra** sulle forme, dove le operazioni $+$ e $\cdot$ corrispondono rispettivamente a unione e intersezione set-teoriche (considerando forme come regioni dello spazio).

Tuttavia, per catturare aspetti metrici e geometrici più ricchi, sono necessarie estensioni. Una **metric shape algebra** incorpora una funzione di distanza $d: S \times S \rightarrow \mathbb{R}^+ \cup \{0\}$ che misura la dissimilarità tra forme. Questa metrica dovrebbe soddisfare:

- $d(a, b) = 0$ se e solo se $a = b$ (identità degli indiscernibili)
- $d(a, b) = d(b, a)$ (simmetria)
- $d(a, c) \leq d(a, b) + d(b, c)$ (disuguaglianza triangolare)

Metriche comuni sulle forme includono la **Hausdorff distance**, che misura la massima distanza da qualsiasi punto di una forma al punto più vicino dell'altra forma, e la **Fréchet distance**, che considera corrispondenze continue tra i confini delle forme.

Proprietà Qualitative Fondamentali: Convessità

La convessità è una delle proprietà geometriche più fondamentali e computazionalmente trattabili. Una regione $R \subset \mathbb{R}^n$ è **convessa** se per ogni coppia di punti $p, q \in R$, l'intero segmento che li connette è contenuto in R :

$$\forall p, q \in R, \forall \lambda \in [0,1]: \lambda p + (1-\lambda)q \in R$$

La convessità ha implicazioni profonde per il ragionamento spaziale e la computazione:

1. **Chiusura sotto Intersezione:** L'intersezione di regioni convesse è convessa, permettendo la rappresentazione di regioni complesse come intersezione di vincoli convessi.

2. **Unicità del Baricentro:** Una regione convessa ha un unico centroide ben definito che appartiene alla regione.
3. **Visibilità:** In una regione convessa, ogni punto è visibile da ogni altro punto (il segmento che li connette non interseca il complemento).
4. **Ottimizzazione:** Problemi di ottimizzazione su regioni convesse hanno struttura computazionale favorevole (unico minimo locale che è anche globale).

Per regioni non convesse, la misura di "quanto" una forma devia dalla convessità è catturata dall'**indice di convessità**:

$$I_{\text{conv}}(R) = \text{Area}(R) / \text{Area}(\text{ConvexHull}(R))$$

dove $\text{ConvexHull}(R)$ è il guscio convesso di R , la più piccola regione convessa che contiene R . Questo indice varia da 0 (regione estremamente concava o disconnessa) a 1 (regione perfettamente convessa).

Una rappresentazione logica della convessità può utilizzare un predicato monadico $\text{Conv}(x)$ che afferma "la regione x è convessa". Assiomi che catturano le proprietà della convessità includono:

Assioma di Intersezione: $\text{Conv}(x) \wedge \text{Conv}(y) \rightarrow \text{Conv}(x \cap y)$

Assioma di Linearità: $\text{Conv}(x) \wedge \text{Conv}(y) \rightarrow \text{Conv}(\lambda x + (1-\lambda)y)$ per $\lambda \in [0,1]$

dove $\lambda x + (1-\lambda)y$ denota la combinazione convessa (Minkowski) delle regioni.

Simmetria: Gruppi e Invarianti

La simmetria è un'altra proprietà fondamentale delle forme, profondamente connessa alla teoria dei gruppi. Una forma possiede simmetria se esiste una trasformazione geometrica non-triviale che la mappa su se stessa.

Il **gruppo di simmetria** di una forma F è l'insieme di tutte le isometrie (trasformazioni che preservano le distanze) che lasciano F invariata:

$$\text{Sym}(F) = \{T: \mathbb{R}^n \rightarrow \mathbb{R}^n \text{ isometria} : T(F) = F\}$$

Questo è un gruppo sotto composizione di trasformazioni. La struttura del gruppo di simmetria classifica le forme:

- **Simmetria ciclica C_n :** Rotazioni di $360^\circ/n$ attorno a un centro
- **Simmetria diedrale D_n :** Rotazioni C_n più n riflessioni
- **Simmetria continua $SO(2)$:** Tutte le rotazioni attorno a un centro (circoli)
- **Simmetria completa $O(2)$:** Tutte le rotazioni e riflessioni (circoli con simmetria radiale completa)

Per poligoni regolari, il gruppo di simmetria è D_n dove n è il numero di lati. Un quadrato ha gruppo D_4 con 8 elementi (4 rotazioni e 4 riflessioni). Un cerchio ha gruppo $O(2)$ infinito.

La simmetria può essere espressa logicamente mediante predicati parametrizzati:

$\text{SymRot}(x, \theta)$: "la forma x ha simmetria rotazionale di angolo θ " $\text{SymRefl}(x, l)$: "la forma x ha simmetria riflessiva rispetto alla linea l "

Assiomi catturano relazioni tra simmetrie:

Composizione di Rotazioni: $\text{SymRot}(x, \theta_1) \wedge \text{SymRot}(x, \theta_2) \rightarrow \text{SymRot}(x, \theta_1 + \theta_2)$

Riflession Double: $\text{SymRefl}(x, l) \rightarrow \text{SymRot}(x, \pi)$ (due riflessioni consecutive danno una rotazione)

Morfologia Matematica: Dilatazione ed Erosione

La morfologia matematica, sviluppata da Georges Matheron e Jean Serra, fornisce un framework algebrico per analizzare e trasformare forme basato su operazioni di dilatazione ed erosione. Questi operatori sono fondamentali nell'elaborazione di immagini e nell'analisi delle forme.

Dato un insieme (forma) $A \subset \mathbb{R}^n$ e un elemento strutturante $B \subset \mathbb{R}^n$ (tipicamente un insieme convesso piccolo), la **dilatazione** di A da B è:

$$A \oplus B = \{z \in \mathbb{R}^n : B_z \cap A \neq \emptyset\} = \bigcup_{a \in A} B_a$$

dove $B_z = \{b + z : b \in B\}$ è la traslazione di B nel punto z .

Intuitivamente, la dilatazione "espande" A , aggiungendo un margine di spessore determinato da B attorno ad A . Se B è un disco di raggio r , la dilatazione aggiunge un margine di larghezza r attorno ad A .

Dualmente, l'**erosione** di A da B è:

$$A \ominus B = \{z \in \mathbb{R}^n : B_z \subseteq A\} = \bigcap_{b \in B} A_{\{-b\}}$$

L'erosione "restringe" A , rimuovendo un margine attorno al confine. Punti vicini al confine vengono eliminati se B centrato in quel punto esce da A .

Proprietà algebriche fondamentali:

Dilatazione:

- Commutativa: $A \oplus B = B \oplus A$
- Associativa: $(A \oplus B) \oplus C = A \oplus (B \oplus C)$
- Distributiva rispetto all'unione: $A \oplus (B \cup C) = (A \oplus B) \cup (A \oplus C)$

Erosione:

- Non commutativa: $A \ominus B \neq B \ominus A$ in generale
- Distributiva rispetto all'intersezione: $A \ominus (B \cup C) = (A \ominus B) \cap (A \ominus C)$

Dualità: $(A \ominus B)^c = A^c \oplus \hat{B}$ dove $\hat{B} = \{-b : b \in B\}$ è il riflesso di B

L'**apertura** e la **chiusura** sono composizioni di questi operatori:

- Apertura: $A \circ B = (A \ominus B) \oplus B$
- Chiusura: $A \bullet B = (A \oplus B) \ominus B$

L'apertura rimuove piccole protrusioni e rumore, mentre la chiusura riempie piccoli buchi e gaps. Entrambe sono idempotenti: $(A \circ B) \circ B = A \circ B$.

Logiche Modali per Operazioni Morfologiche

La morfologia matematica può essere formalizzata logicamente mediante logiche modali dove operatori morfologici diventano modalità logiche.

Definiamo operatori modali $\Diamond_B$ e $\Box_B$ parametrizzati dall'elemento strutturante B :

$\Diamond_B \phi$: "esiste un punto nella dilatazione di ϕ da B " $\Box_B \phi$: "tutti i punti nell'erosione di ϕ da B soddisfano ϕ "

La semantica è definita su modelli spaziali $M = (R^n, V)$ dove V assegna a ogni formula proposizionale p un sottoinsieme di R^n (la regione dove p è vera):

$M, x \models \Diamond_B \phi$ se e solo se $B_x \cap [[\phi]]_M \neq \emptyset$, $M, x \models \Box_B \phi$ se e solo se $B_x \subseteq [[\phi]]_M$

dove $[[\phi]]_M = \{y \in R^n : M, y \models \phi\}$.

Assiomi che catturano proprietà morfologiche:

Assioma K Morfologico: $\Box_B(\phi \rightarrow \psi) \rightarrow (\Box_B \phi \rightarrow \Box_B \psi)$

Assioma di Dualità: $\neg \Diamond_B \neg \phi \leftrightarrow \Box_B \phi$

Assioma di Distribuzione: $\Diamond_B (\phi \vee \psi) \leftrightarrow (\Diamond_B \phi \vee \Diamond_B \psi)$

Assioma di Idempotenza per Apertura: $\Box_B \Diamond_B \phi \rightarrow \Box_B \Diamond_B \Box_B \Diamond_B \phi$

Questi assiomi permettono di ragionare formalmente su trasformazioni morfologiche e loro composizioni senza calcolare esplicitamente le geometrie.

Descrittori di Forma: Momenti e Invarianti

Per il riconoscimento e la classificazione di forme, sono necessari descrittori numerici che catturano caratteristiche essenziali in modo compatto. I **momenti** geometrici forniscono una famiglia di descrittori basati su integrali pesati sulla regione.

Per una regione $R \subset \mathbb{R}^2$ con funzione caratteristica $\chi_R(x,y)$, il momento di ordine $(p+q)$ è:

$$M_{pq} = \iint_R x^p y^q dx dy$$

I momenti di ordine basso hanno interpretazioni geometriche:

- M_{00} : Area della regione
- $(M_{10}/M_{00}, M_{01}/M_{00})$: Coordinate del centroide
- M_{20}, M_{02}, M_{11} : Momenti di secondo ordine, correlati a orientamento e estensione

I **momenti centrali** μ_{pq} sono calcolati rispetto al centroide, rimuovendo dipendenza dalla posizione:

$$\mu_{pq} = \iint_R (x - \bar{x})^p (y - \bar{y})^q dx dy$$

dove $(\bar{x}, \bar{y})$ è il centroide.

I **momenti invarianti di Hu** sono combinazioni di momenti centrali normalizzati che sono invarianti rispetto a traslazione, rotazione, e scaling. I primi due invarianti di Hu sono:

$$\varphi_1 = \eta_{20} + \eta_{02}$$

$$\varphi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$

dove $\eta_{pq} = \mu_{pq} / \mu^{((p+q)/2+1)}_{00}$ sono i momenti centrali normalizzati.

Questi invarianti permettono il riconoscimento di forme indipendentemente da trasformazioni di similitudine, fondamentale per sistemi di visione computerizzata.

Scheletri e Medial Axis

Lo **scheletro** o **medial axis** di una forma è una rappresentazione intrinseca che cattura la "spina dorsale" topologica della forma. Per una regione R , il medial axis $MA(R)$ è definito come:

$$MA(R) = \{x \in R : \text{esistono almeno due punti distinti } p, q \in \partial R \text{ tali che } d(x,p) = d(x,q) = d(x, \partial R)\}$$

dove ∂R è il confine di R e $d(x, \partial R) = \inf\{d(x,y) : y \in \partial R\}$ è la distanza da x al confine.

Intuitivamente, il medial axis è il luogo dei centri di cerchi massimali contenuti in R (cerchi che toccano il confine in almeno due punti).

Proprietà fondamentali del medial axis:

1. **Homotopy Equivalent:** $MA(R)$ ha lo stesso tipo di omotopia di R (stessa connettività topologica)

2. **Dimensione Ridotta:** Per regioni 2D, $MA(R)$ è tipicamente un grafo 1D (curve)
3. **Ricostruibilità:** R può essere completamente ricostruita da $MA(R)$ insieme alla funzione raggio $r: MA(R) \rightarrow R^+$ che assegna a ogni punto del medial axis il raggio del cerchio massimale centrato lì

La ricostruzione è data da:

$$R = \bigcup \{x \in MA(R)\} B\{r(x)\}(x)$$

dove $B_r(x)$ è il disco di raggio r centrato in x .

Il medial axis è computazionalmente complesso da calcolare esattamente (richiede costruzioni di diagrammi di Voronoi generalizzati), ma approssimazioni efficienti esistono basate su thinning morfologico o grassfire transforms.

Fourier Descriptors per Contorni

Per forme descritte dal loro contorno (curve chiuse nel piano), i **Fourier descriptors** forniscono una rappresentazione in frequenza compatta e informativa.

Un contorno chiuso può essere parametrizzato come una curva complessa $z(t) = x(t) + i \cdot y(t)$ per $t \in [0, 1]$. La serie di Fourier discreta dei punti del contorno campionati è:

$$a_k = (1/N) \sum_{n=0}^{N-1} z_n \exp(-i 2\pi kn/N)$$

dove z_n sono N punti campionati uniformemente sul contorno.

I coefficienti $\{a_k\}$ sono i Fourier descriptors. Le prime componenti (basse frequenze) catturano la forma globale, mentre componenti di frequenza alta catturano dettagli fini. Troncando la serie a $M < N$ coefficienti, si ottiene una rappresentazione approssimata ma compatta:

$$z'(t) \approx \sum_{k=-M/2}^{M/2} a_k \exp(i 2\pi kt)$$

Invarianza: Trasformazioni geometriche del contorno corrispondono a trasformazioni semplici dei descrittori:

- Traslazione: a_0 cambia, a_k ($k \neq 0$) invariati
- Rotazione di θ : $a_k \rightarrow a_k \exp(i k\theta)$
- Scaling di s : $a_k \rightarrow s \cdot a_k$
- Punto di partenza: $a_k \rightarrow a_k \exp(i 2\pi k \cdot \text{shift})$

Descrittori completamente invarianti si ottengono normalizzando:

$$\phi_k = |a_k| / |a_1| \quad (k \geq 2)$$

che sono invarianti rispetto a traslazione, rotazione, scaling, e punto di partenza.

Rappresentazioni Multi-Risoluzione: Wavelets e Piramidi

Per catturare forme a diverse scale, rappresentazioni multi-risoluzione sono essenziali. Le **piramidi di forme** decompongono una forma in versioni progressivamente semplificate ottenute tramite smoothing e subsampling.

Una piramide Gaussiana applica iterativamente un filtro Gaussiano seguito da decimazione:

$$L_0 = \text{originale} \quad L_i = \text{downsample}(\text{Gaussian_blur}(L_{i-1}))$$

La piramide Laplaciana, derivata dalla Gaussiana, rappresenta differenze tra livelli consecutivi, catturando informazioni a diverse scale:

$$\text{Lap}_i = L_i - \text{upsample}(L_{i+1})$$

Le **wavelet transforms** generalizzano questo approccio fornendo una base matematica rigorosa per decomposizione multi-risoluzione. Una wavelet madre $\psi(t)$ genera una famiglia di funzioni per dilatazione e traslazione:

$$\psi_{\{a,b\}}(t) = (1/\sqrt{a}) \psi((t-b)/a)$$

La trasformata wavelet continua di una funzione $f(t)$ è:

$$W_f(a,b) = \int f(t) \psi_{\{a,b\}}^*(t) dt$$

Per contorni di forme, wavelet transforms catturano caratteristiche locali a diverse scale, permettendo analisi gerarchica e multi-risoluzione.

Shape Grammars e Sistemi Generativi

Le **shape grammars**, introdotte da Stiny e Gips, forniscono un formalismo generativo per descrivere e generare famiglie di forme attraverso regole di produzione.

Una shape grammar è una quadrupla $G = (V_T, V_N, R, I)$ dove:

- V_T è un insieme di forme terminali (primitive)
- V_N è un insieme di forme non-terminali (simboli)
- R è un insieme di regole di produzione $\alpha \rightarrow \beta$
- I è una forma iniziale (assioma)

Una regola $\alpha \rightarrow \beta$ specifica che la sottoforma α può essere sostituita dalla sottoforma β , potenzialmente con trasformazioni geometriche.

Esempio: Grammar per generare alberi frattali:

- Terminali: segmento di linea L
- Non-terminali: simbolo A (ramo)
- Regole:

- $A \rightarrow L + [A] - [A]$ (un ramo si biforca in due sub-rami)
- $A \rightarrow L$ (un ramo termina)
- Iniziale: A

Applicando iterativamente le regole, si generano strutture arborescenti auto-simili.

Shape grammars sono potenti per:

- Design parametrico e generativo in architettura
- Sintesi procedurale di contenuti grafici
- Modellazione di patterns ricorrenti in forme naturali e artistiche

Logiche Descrittive per Forme Complesse

Per esprimere vincoli complessi su forme e ragionare su gerarchie di forme, le logiche descrittive (Description Logics) possono essere estese con predicati spaziali e geometrici.

Un'ontologia di forme in OWL (Web Ontology Language, basato su Description Logic) potrebbe includere:

Classi:

- Shape (classe generale delle forme)
 - Polygon $\sqsubseteq$ Shape
 - Triangle $\sqsubseteq$ Polygon
 - Quadrilateral $\sqsubseteq$ Polygon
 - Rectangle $\sqsubseteq$ Quadrilateral
 - Square $\sqsubseteq$ Rectangle
 - Curve $\sqsubseteq$ Shape
 - Circle $\sqsubseteq$ Curve
 - Ellipse $\sqsubseteq$ Curve

Proprietà:

- hasSymmetry: Shape $\rightarrow$ SymmetryType
- hasConvexity: Shape $\rightarrow$ xsd:float [0,1]
- hasSideCount: Polygon $\rightarrow$ xsd:integer
- encloses: Shape $\rightarrow$ Shape
- adjacent: Shape $\rightarrow$ Shape

Assiomi:

- Square $\equiv$ Rectangle $\sqcap$ (hasSideCount exactly 4) $\sqcap$ (hasAllSidesEqual value true)
- ConvexShape $\equiv$ Shape $\sqcap$ (hasConvexity value 1.0)
- RegularPolygon $\equiv$ Polygon $\sqcap$ (hasAllSidesEqual value true) $\sqcap$ (hasAllAnglesEqual value true)

Reasoner automatici possono inferire appartenenze a classi, verificare consistenza, e rispondere a query complesse:

Query: "Trova tutte le forme convesse che racchiudono almeno tre cerchi"

Questa query può essere tradotta in una descrizione logica:

ConvexShape $\sqsupseteq$ (≥ 3 encloses.Circle)

Il reasoner restituisce tutte le forme nell'ontologia che soddisfano questa descrizione.

Applicazioni e Prospettive Future

Le logiche per forme trovano applicazioni in:

Computer Graphics: Modellazione procedurale, riconoscimento di pattern, compressione di modelli geometrici

Computer Vision: Riconoscimento di oggetti indipendente da viewpoint, segmentazione basata su forma, tracking di oggetti deformabili

GIS: Analisi di forme di regioni geografiche, classificazione di land cover, urban planning

Biologia Computazionale: Analisi morfologica di cellule e organismi, classificazione tassonomica basata su morfologia

Design Assistito: Generazione automatica di varianti di design, ottimizzazione di forme sotto vincoli funzionali, verifica di requisiti geometrici

Le direzioni future includono l'integrazione di deep learning per apprendere rappresentazioni di forme da dati, lo sviluppo di logiche ancora più espressive per catturare vincoli geometrici complessi, e l'applicazione a domini emergenti come la stampa 3D personalizzata e la biologia sintetica.

Incertezza Spaziale Avanzata

Fondamenti dell'Incertezza nella Rappresentazione Spaziale

L'incertezza è un aspetto intrinseco e inevitabile della rappresentazione e del ragionamento spaziale. Le fonti di incertezza sono molteplici e pervasive: imprecisione nei sensori e strumenti di misurazione, incompletezza dei dati disponibili, vaghezza intrinseca dei concetti spaziali naturali (cos'è esattamente una "montagna" o una "collina"?), variabilità temporale dei fenomeni spaziali, e limitazioni cognitive nella percezione e descrizione dello spazio.

La rappresentazione formale dell'incertezza spaziale richiede framework matematici che vadano oltre la logica classica bivalente e la geometria deterministica. Tre paradigmi principali si sono affermati per trattare diverse forme di incertezza: la **Rough Set Theory** per l'indeterminatezza dovuta a granularità limitata dell'informazione, la **Fuzzy Logic** per la

vaghezza e gradualità dei concetti spaziali, e la **Possibility Theory** per l'incertezza qualitativa non-probabilistica.

Questi approcci non sono mutuamente esclusivi ma complementari, ciascuno adatto a catturare aspetti specifici dell'incertezza. La scelta del framework dipende dalla natura epistemologica dell'incertezza: è dovuta a limiti nell'osservazione (rough sets), a confini intrinsecamente vaghi (fuzzy logic), o a mancanza di informazione completa sullo stato del mondo (possibility theory)?

Rough Set Theory Applicata a Regioni con Confini Vaghi

La Rough Set Theory, sviluppata da Zdzisław Pawlak negli anni Ottanta, fornisce un framework matematico per trattare l'indeterminatezza derivante da classificazioni imperfette o granularità limitata dell'informazione. Nel contesto spaziale, questa teoria è particolarmente appropriata per rappresentare regioni i cui confini non possono essere determinati con precisione data la risoluzione disponibile dei dati.

Il concetto centrale è quello di **approssimazione**: data una relazione di indiscernibilità R su un universo U (che nel caso spaziale potrebbe essere l'insieme di tutte le celle di una griglia di misurazione), qualsiasi sottoinsieme $X \subseteq U$ (una regione spaziale) può essere approssimato da due insiemi:

La **lower approximation** (approssimazione inferiore) $R_*(X)$:

$$R_*(X) = \{x \in U : [x]_R \subseteq X\}$$

dove $[x]_R = \{y \in U : xRy\}$ è la classe di equivalenza di x rispetto a R . La lower approximation contiene tutti gli elementi che sicuramente appartengono a X dato il nostro livello di granularità.

La **upper approximation** (approssimazione superiore) $R^*(X)$:

$$R^*(X) = \{x \in U : [x]_R \cap X \neq \emptyset\}$$

che contiene tutti gli elementi che possibilmente appartengono a X .

La **boundary region** (regione di confine) è:

$$BN_R(X) = R^*(X) \setminus R_*(X)$$

che contiene elementi per cui non possiamo determinare con certezza l'appartenenza a X data la granularità dell'informazione.

Un insieme X è detto **R-definito** (preciso) se $R_*(X) = R^*(X)$, cioè se non ha boundary region. Un insieme è **rough** se $BN_R(X) \neq \emptyset$.

Esempio Spaziale: Consideriamo una regione geografica "montagna" e un sistema di misurazione che classifica celle di una griglia in base all'altitudine con granularità di 100

metri. Due celle sono indiscernibili se hanno la stessa classe di altitudine. La regione "montagna" (definita come altitudine > 1500m) sarà rough se ci sono celle nella classe 1400-1500m o 1500-1600m che contengono territorio sia sopra che sotto 1500m.

Misure di Roughness

La "rugosità" di un insieme può essere quantificata mediante l'**accuracy** (precisione):

$$\alpha_R(X) = |R_-(X)| / |R(X)|$$

dove $|\cdot|$ denota la cardinalità. Questo valore varia da 0 (massima imprecisione) a 1 (insieme preciso).

Un'altra misura è il **roughness degree**:

$$\rho_R(X) = 1 - \alpha_R(X) = |BN_R(X)| / |R^*(X)|$$

Operazioni su Rough Sets Spaziali

Le operazioni standard set-teoriche si estendono ai rough sets operando separatamente su lower e upper approximations:

Complemento:

- $R_-(X^c) = (R(X))^c$
- $R^*(X^c) = (R_-(X))^c$

Unione:

- $R_-(X \cup Y) \supseteq R_-(X) \cup R_-(Y)$
- $R^*(X \cup Y) = R^*(X) \cup R^*(Y)$

Intersezione:

- $R_-(X \cap Y) = R_-(X) \cap R_-(Y)$
- $R^*(X \cap Y) \subseteq R^*(X) \cap R^*(Y)$

Queste proprietà permettono di ragionare su operazioni spaziali (unione di regioni, intersezione di aree, complemento) preservando la struttura dell'incertezza.

Relazioni Topologiche Rough

Le relazioni topologiche tra regioni spaziali rough sono più complesse che nel caso deterministico. Per due rough sets A e B, possiamo definire versioni "possibly" e "certainly" delle relazioni RCC-8:

Certainly Disconnected: $C(A \cap B) \equiv R_-(A) \cap R_-(B) = \emptyset$

Possibly Connected: $P(A \cap B) \equiv R^*(A) \cap R^*(B) \neq \emptyset$

Analogamente per altre relazioni. Questo genera una gerarchia di 64 possibili relazioni "certain-possible" tra rough regions, molte delle quali però collassano o sono equivalenti sotto vincoli ragionevoli.

Fuzzy Spatial Logic: Fondamenti e Estensioni

La Fuzzy Logic, introdotta da Lotfi Zadeh, tratta la vaghezza attraverso funzioni di appartenenza che assegnano gradi di membership nell'intervallo $[0,1]$ piuttosto che valori booleani. Nel contesto spaziale, questo è fondamentale per rappresentare concetti con confini intrinsecamente vaghi come "vicino", "lontano", "grande", "montagna", "pianura".

Un **fuzzy spatial set** A in un universo spaziale X è caratterizzato da una funzione di membership:

$$\mu_A: X \rightarrow [0,1]$$

dove $\mu_A(x)$ rappresenta il grado in cui il punto (o la cella) x appartiene all'insieme A . Valori estremi 0 e 1 indicano non-appartenenza certa e appartenenza certa, mentre valori intermedi catturano transizioni graduali.

Operazioni Fuzzy Standard

Le operazioni set-teoriche classiche sono estese al caso fuzzy:

Complemento (T-complement): $\mu_{A^c}(x) = 1 - \mu_A(x)$

Intersezione (T-norm): $\mu_{\{A \cap B\}}(x) = T(\mu_A(x), \mu_B(x))$

dove T è una t-norm, tipicamente:

- Minimum (Zadeh): $T(a,b) = \min(a,b)$
- Product: $T(a,b) = a \cdot b$
- Łukasiewicz: $T(a,b) = \max(0, a+b-1)$

Unione (T-conorm): $\mu_{\{A \cup B\}}(x) = S(\mu_A(x), \mu_B(x))$

dove S è una t-conorm, tipicamente:

- Maximum (Zadeh): $S(a,b) = \max(a,b)$
- Probabilistic sum: $S(a,b) = a+b-a \cdot b$
- Łukasiewicz: $S(a,b) = \min(1, a+b)$

Relazioni Topologiche Fuzzy

Estendere relazioni topologiche al caso fuzzy richiede di fuzzificare il concetto di intersezione e confine. Per due fuzzy regions A e B , la connessione può essere definita come:

$$C_{\text{fuzzy}}(A, B) = \sup_{\{x \in X\}} \min(\mu_A(x), \mu_B(x))$$

che misura il grado massimo di sovrapposizione tra le due regioni fuzzy.

Il RCC-8 fuzzy può essere definito mediante soglie:

- **Certainly Disconnected:** $C_fuzzy(A, B) = 0$
- **Possibly Connected:** $C_fuzzy(A, B) > 0$
- **Highly Connected:** $C_fuzzy(A, B) > \theta$ per qualche soglia θ

Approcci più sofisticati utilizzano egg-yolk representations dove ogni regione fuzzy è rappresentata da un "tuorlo" (core, dove $\mu = 1$) e un "albume" (support, dove $\mu > 0$), permettendo di definire relazioni tra i core e tra i support.

Distanza Fuzzy

La distanza tra regioni fuzzy non è un singolo valore ma può essere anch'essa fuzzy. Diversi approcci esistono:

Distanza di Hausdorff Fuzzy: Estende la distanza di Hausdorff considerando α -cuts:

$$d_H(A, B) = \sup_{\alpha \in (0,1]} d_H(A_\alpha, B_\alpha)$$

dove $A_\alpha = \{x : \mu_A(x) \geq \alpha\}$ è l' α -cut di A.

Distanza Media Ponderata:

$$d_avg(A, B) = (\iint \mu_A(x) \mu_B(y) ||x-y|| dx dy) / (\iint \mu_A(x) \mu_B(y) dx dy)$$

che media la distanza euclidea pesata dalle funzioni di membership.

Direzioni Fuzzy

Le direzioni qualitative possono essere fuzzificate per catturare transizioni graduali tra categorie. Invece di partizioni nette dello spazio (nord, nord-est, est, ...), utilizziamo funzioni di membership trapezoidal o Gaussiane:

$$\mu_North(\theta) = \exp(-(\theta - 90^\circ)^2 / (2\sigma^2))$$

dove θ è l'angolo e σ controlla la "larghezza" della categoria nord.

Un punto B può quindi essere "a nord di" A con grado 0.8 e "a nord-est di" A con grado 0.3 simultaneamente, riflettendo la natura graduale delle categorie direzionali.

Logiche Fuzzy Multi-Valued

La formalizzazione logica del ragionamento fuzzy utilizza logiche multi-valued dove le formule hanno valori di verità nell'intervallo $[0,1]$.

Il linguaggio include operatori fuzzy:

$$\varphi ::= p \mid \neg \varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \rightarrow \psi$$

La semantica è data da valutazioni $V: \text{Formulas} \rightarrow [0,1]$ che soddisfano:

$V(\neg\varphi) = 1 - V(\varphi)$ $V(\varphi \wedge \psi) = T(V(\varphi), V(\psi))$ per qualche t-norm T $V(\varphi \vee \psi) = S(V(\varphi), V(\psi))$ per qualche t-conorm S $V(\varphi \rightarrow \psi) = S(1 - V(\varphi), V(\psi))$ (implicazione di Łukasiewicz)

o alternativamente: $V(\varphi \rightarrow \psi) = T(V(\varphi), V(\psi)) / V(\varphi)$ se $V(\varphi) > 0$, altrimenti 1

Regole di inferenza operano su gradi di verità:

Modus Ponens Fuzzy: Se $V(\varphi) = a$ e $V(\varphi \rightarrow \psi) = b$, allora $V(\psi) \geq T(a, b)$

Questo permette di derivare conclusioni con gradi di certezza appropriati.

Possibility Theory per Incertezza Qualitativa

La Possibility Theory, sviluppata da Zadeh e Dubois-Prade, fornisce un framework per ragionare su incertezza qualitativa che non richiede probabilità precise. È particolarmente adatta quando l'informazione disponibile è troppo imprecisa per assegnare probabilità ma permette di distinguere tra stati più o meno plausibili.

Una **distribuzione di possibilità** $\pi: \Omega \rightarrow [0,1]$ assegna a ogni stato $\omega \in \Omega$ un grado di possibilità, con la convenzione che $\max_{\{\omega \in \Omega\}} \pi(\omega) = 1$ (almeno uno stato è completamente possibile).

Per un evento (insieme di stati) $A \subseteq \Omega$, definiamo:

Possibilità: $\Pi(A) = \sup_{\{\omega \in A\}} \pi(\omega)$

misura quanto è possibile che A si verifichi (il massimo grado di possibilità di stati in A).

Necessità: $N(A) = 1 - \Pi(A^c) = \inf_{\{\omega \notin A\}} (1 - \pi(\omega))$

misura quanto è certo che A si verifichi (il minimo grado di impossibilità di stati fuori da A).

Proprietà fondamentali:

- $\Pi(\emptyset) = 0$, $\Pi(\Omega) = 1$
- $\Pi(A \cup B) = \max(\Pi(A), \Pi(B))$
- $N(A \cap B) = \min(N(A), N(B))$
- $N(A) \leq \Pi(A)$ (se qualcosa è necessario, è anche possibile)
- $N(A) > 0$ implica $\Pi(A) = 1$

Possibility Theory per Regioni Spaziali

Nel contesto spaziale, la possibility theory può modellare incertezza sulla localizzazione di entità. Supponiamo di cercare un oggetto in una regione, ma abbiamo solo informazione imprecisa sulla sua posizione.

Una **distribuzione di possibilità spaziale** π : $\text{Space} \rightarrow [0,1]$ assegna a ogni locazione un grado di possibilità che l'oggetto si trovi lì.

Esempio: "L'oggetto è vicino alla torre ma non troppo lontano dal lago"

Possiamo definire:

- $\pi_{\text{tower}}(x) = 1 - (d(x, \text{tower}) / d_{\text{max}})$ (possibilità decresce con distanza dalla torre)
- $\pi_{\text{lake}}(x)$ = funzione che è alta vicino al lago e decresce

La distribuzione combinata usando un operatore aggregante (min per congiunzione, max per disgiunzione):

$$\pi_{\text{combined}}(x) = \min(\pi_{\text{tower}}(x), \pi_{\text{lake}}(x))$$

Relazioni Spaziali Possibilistiche

Le relazioni spaziali possono essere espresse possibilisticamente:

$\Pi(\text{DC}(A, B))$: Possibilità che A e B siano disconnesse $N(\text{DC}(A, B))$: Necessità che A e B siano disconnesse

Dato un insieme di vincoli possibilistici su relazioni:

$$\{(\Pi(R_i(A_i, B_i)) \geq \alpha_i) : i = 1, \dots, n\}$$

il problema di consistenza chiede se esiste una configurazione spaziale compatibile con tutti i vincoli.

La propagazione di vincoli possibilistici utilizza regole:

Regola del Min per Congiunzione: Se $\Pi(R_1) \geq \alpha$ e $\Pi(R_2) \geq \beta$, allora $\Pi(R_1 \wedge R_2) \geq \min(\alpha, \beta)$

Regola della Composizione: Se $\Pi(R(A,B)) \geq \alpha$ e $\Pi(S(B,C)) \geq \beta$, allora $\Pi(R \circ S(A,C)) \geq \min(\alpha, \beta)$

dove $R \circ S$ è la composizione delle relazioni.

Confronto tra Probability e Possibility

Possibilità e probabilità sono misure complementari ma distinte:

- **Additività:** $P(A) + P(A^c) = 1$ (probabilità), ma $\Pi(A) + \Pi(A^c) \geq 1$ (possibilità)
- **Informazione:** Probabilità richiede informazione quantitativa precisa; possibilità opera con informazione qualitativa
- **Max-additività:** $\Pi(A \cup B) = \max(\Pi(A), \Pi(B))$; $P(A \cup B) = P(A) + P(B) - P(A \cap B)$

La transformation tra probabilità e possibilità non è unica, ma una corrispondenza ragionevole è:

Se $P(\omega)$ è conosciuta, possiamo definire: $\pi(\omega) = 1$ se $\omega \in \arg \max_{\omega'} P(\omega')$ $\pi(\omega) < 1$ altrimenti, calibrato in modo che rifletta la "sorpresa"

Inversamente, da π possiamo ottenere una famiglia di distribuzioni di probabilità compatibili:

P compatibile con π se $P(\{\omega : \pi(\omega) \geq \alpha\}) \geq \Pi^*(\alpha)$ per tutti α

dove Π^* è una trasformazione appropriata.

Integrazione Multi-Teoria: Rough-Fuzzy Sets

Le teorie di rough sets e fuzzy sets possono essere integrate per catturare sia indeterminatezza (rough) che vaghezza (fuzzy) simultaneamente.

Un **rough fuzzy set** è una coppia $(R_-(\mu_A), R_+(\mu_A))$ dove μ_A è una funzione di membership fuzzy e R_- , R_+ sono operatori di approssimazione rough applicati alle funzioni fuzzy.

Per ogni x , definiamo:

$$R_-(\mu_A)(x) = \inf_{y \in [x]_R} \mu_A(y)$$

$$R_+(\mu_A)(x) = \sup_{y \in [x]_R} \mu_A(y)$$

La lower approximation fornisce il grado minimo di appartenenza fuzzy nella classe di equivalenza di x , mentre la upper approximation fornisce il massimo.

Un **fuzzy rough set** è l'approccio duale: applichiamo relazioni di indiscernibilità fuzzy a insiemi crisp o fuzzy.

La relazione di indiscernibilità è fuzzificata: $R(x,y) \in [0,1]$ rappresenta il grado in cui x e y sono indiscernibili.

Le approssimazioni diventano:

$$R_-(A)(x) = \inf_{y \in U} \max(1 - R(x,y), \chi_A(y))$$

$$R_+(A)(x) = \sup_{y \in U} \min(R(x,y), \chi_A(y))$$

Questa integrazione è particolarmente potente per rappresentare fenomeni spaziali che esibiscono sia confini vaghi (fumoso) che risoluzione limitata (rough).

Applicazioni Pratiche

Le tecniche di incertezza spaziale trovano applicazioni in:

Land Cover Classification: Pixel in immagini satellitari possono appartenere a multiple classi (foresta, prateria, urbano) con gradi di membership fuzzy, e confini tra classi sono rough a causa della risoluzione limitata.

Modellazione di Rischio Naturale: Zone a rischio di alluvione o terremoto hanno confini intrinsecamente vaghi. Possibility theory permette di esprimere "è possibile ma non necessario che questa area sia allagata".

GIS Integrato con Conoscenza Incerta: Database spaziali che integrano dati precisi (coordinate GPS) con descrizioni vaghe ("vicino al centro", "nella parte nord della città") richiedono rappresentazioni fuzzy e possibilistiche.

Robotica Mobile in Ambienti Incerti: Mappe occupancy grid possono usare probabilità, possibilità, o fuzzy membership per rappresentare incertezza sulla presenza di ostacoli in celle della griglia.

Analisi Archeologica: Ricostruzioni di siti archeologici da evidenze frammentarie usano rough sets per rappresentare indeterminazione e possibility theory per esprimere ipotesi alternative.

Direzioni Future

Le sfide aperte includono:

- Sviluppo di reasoning engines efficienti che integrano multiple teorie di incertezza
- Apprendimento automatico di funzioni di membership fuzzy e distribuzioni di possibilità da dati
- Estensione a scenari dinamici dove l'incertezza evolve nel tempo
- Interfacce utente intuitive per specificare e visualizzare incertezza spaziale multi-forma
- Standardizzazione di formati di dati e ontologie per incertezza spaziale interoperabile

L'obiettivo ultimo è creare sistemi che gestiscano l'incertezza in modo robusto, trasparente, e cognitivamente plausibile, riflettendo la natura intrinsecamente imprecisa e incompleta della conoscenza spaziale umana e artificiale.

Logiche Multi-Granularità

Fondamenti della Rappresentazione Multi-Scala dello Spazio

La rappresentazione dello spazio a diversi livelli di granularità costituisce una caratteristica fondamentale del ragionamento spaziale sia umano che computazionale. Gli esseri umani navigano naturalmente tra diverse scale di astrazione: possiamo pensare a una città come un singolo punto su una mappa nazionale, come un insieme di quartieri su una mappa urbana, o come un dettagliato network di strade su una mappa locale. Questa capacità di zoom concettuale, di astrazione e raffinamento, è essenziale per gestire la complessità intrinseca dell'informazione spaziale.

Il concetto di granularità spaziale si riferisce al livello di dettaglio con cui rappresentiamo e ragioniamo sullo spazio. Una rappresentazione a granularità grossa tratta ampie regioni come unità atomiche indivisibili, mentre una rappresentazione a granularità fine distingue dettagli minuti all'interno di queste regioni. La scelta della granularità appropriata dipende dal compito cognitivo o computazionale in corso: per pianificare un viaggio internazionale, una granularità nazionale o regionale è appropriata, mentre per parcheggiare un'automobile, è necessaria una granularità metrica precisa.

Le logiche multi-granularità forniscono framework formali per rappresentare conoscenza spaziale a diversi livelli di dettaglio simultaneamente, per stabilire connessioni e corrispondenze tra questi livelli, e per ragionare in modo coerente attraverso le scale. Questo è fondamentale per sistemi di informazione geografica, per la modellazione di fenomeni naturali che operano a scale multiple, per la fusione di dati provenienti da sensori con diverse risoluzioni, e per molte altre applicazioni.

Gerarchie di Granularità: Struttura e Proprietà

Una gerarchia di granularità è una struttura ordinata di livelli di dettaglio, dove ogni livello fornisce una partizione o copertura dello spazio a una certa risoluzione, e i livelli sono connessi da relazioni di raffinamento o astrazione. Formalmente, possiamo concepire una gerarchia di granularità come una sequenza di partizioni $P_0, P_1, P_2, \dots, P_n$, dove P_0 è la partizione più grossa (l'intero spazio come singola unità) e P_n è la partizione più fine (potenzialmente punti individuali o celle minime).

Ogni partizione P_i consiste di un insieme di regioni mutuamente esclusive la cui unione copre l'intero spazio di interesse. La relazione di raffinamento tra livelli consecutivi richiede che ogni regione in P_i sia l'unione di una o più regioni in P_{i+1} . In altre parole, passando a un livello più fine, le regioni possono solo essere suddivise, mai fuse attraverso i confini delle regioni del livello più grosso.

Questa proprietà di consistenza gerarchica è fondamentale per garantire che il reasoning a diversi livelli sia coerente. Se una regione R al livello i si sovrappone con una regione S allo stesso livello, allora tutte le sotto-regioni di R al livello $i+1$ devono intersecare solo sotto-regioni di S , non sotto-regioni di altre regioni al livello i .

Le gerarchie di granularità possono essere uniformi o non uniformi. In una gerarchia uniforme, il fattore di raffinamento è costante: ogni regione al livello i viene suddivisa nello stesso numero di sotto-regioni al livello $i+1$. Per esempio, una griglia quadrata può essere raffinata uniformemente dividendo ogni cella in quattro quadranti, producendo una quadtree. In una gerarchia non uniforme, il fattore di raffinamento può variare: alcune regioni possono essere suddivise finemente mentre altre rimangono relativamente grosse, riflettendo la distribuzione non uniforme della rilevanza o complessità dell'informazione spaziale.

Partizioni Amministrative e Gerarchie Socio-Geografiche

Un esempio naturale di gerarchia di granularità è dato dalle divisioni amministrative di uno stato: nazione, regioni o stati, province o contee, comuni, quartieri o distretti. Ogni livello fornisce una partizione del territorio in unità governative o statistiche, e i livelli sono

strutturati gerarchicamente per definizione: ogni comune appartiene a esattamente una provincia, ogni provincia a esattamente una regione, e così via.

Questa struttura gerarchica permette di aggregare dati e statistiche a diversi livelli. Per esempio, la popolazione totale di una regione è la somma delle popolazioni delle sue province, che a loro volta sono le somme delle popolazioni dei comuni che le compongono. Il reasoning spaziale può sfruttare questa struttura: se sappiamo che un evento è accaduto nel comune A, sappiamo automaticamente che è accaduto nella provincia che contiene A e nella regione che contiene quella provincia.

Tuttavia, le gerarchie amministrative non sempre soddisfano perfettamente i criteri formali di una gerarchia di granularità. Possono esistere entità con status speciale che non si inseriscono pulitamente nella gerarchia (città metropolitane con confini che non coincidono con quelli provinciali, territori autonomi, zone economiche speciali). Inoltre, le divisioni amministrative cambiano nel tempo attraverso riforme territoriali, creando sfide per il reasoning temporale multi-granularità.

Gerarchie Naturali e Watershed Delineations

Nella geografia fisica, molti fenomeni naturali esibiscono strutture gerarchiche. Un esempio importante è la rete idrografica: piccoli rivoli si uniscono a formare ruscelli, che confluiscono in torrenti, che alimentano fiumi, che sfociano in mari o oceani. Questa struttura dendritica può essere vista come una gerarchia di granularità, dove al livello più fine abbiamo singoli segmenti di corsi d'acqua, e ai livelli più grossi abbiamo interi bacini idrografici.

La watershed delineation, la suddivisione del territorio in bacini idrografici gerarchici, è un problema fondamentale in idrologia computazionale. Algoritmi basati su digital elevation models identificano automaticamente le linee di spartiacque che separano bacini, e costruiscono gerarchie di bacini annidati. Ogni bacino al livello $i+1$ che contribuisce allo stesso punto di confluenza viene aggregato in un singolo bacino al livello i .

Il reasoning multi-granularità su watershed è cruciale per modellare il trasporto di inquinanti, la propagazione di inondazioni, e la gestione delle risorse idriche. Per esempio, l'inquinamento che entra in un piccolo affluente si propagherà attraverso la gerarchia fino a influenzare l'intero bacino principale, e la velocità e la diluizione dipendono dalla struttura gerarchica della rete.

Gerarchie Topografiche: Valley e Landform Hierarchies

Il terreno può essere decomposto gerarchicamente in landforms (forme del terreno) di diversa scala: montagne, colline, valli, pianure, creste. Una montagna importante al livello grosso può contenere creste secondarie e valli laterali al livello fine, che a loro volta contengono caratteristiche topografiche ancora più minute.

La segmentazione automatica del terreno in landforms gerarchiche è un problema attivo di ricerca in geomorfologia computazionale. Approcci basati su watershed dal modello di elevazione (invertito per identificare le montagne come "bacini" e le valli come "fiumi") o su analisi spettrale del terreno (frequenze spaziali diverse corrispondono a scale diverse) producono decomposizioni gerarchiche del paesaggio.

Queste gerarchie sono utili per il reasoning qualitativo su relazioni topografiche: una località può essere descritta come "nella valle del fiume X, ai piedi della montagna Y, sul versante nord della cresta Z", dove valle, montagna e cresta sono identificati a diversi livelli della gerarchia topografica. Il livello appropriato di descrizione dipende dal contesto comunicativo e dalla familiarità dell'ascoltatore con la geografia locale.

Relazioni tra Livelli di Granularità

Le connessioni formali tra livelli di granularità sono essenziali per il reasoning coerente attraverso le scale. Esistono diversi tipi di relazioni inter-livello, ciascuna con proprietà e utilizzi specifici.

Relazioni di Raffinamento e Astrazione

La relazione fondamentale è quella di raffinamento (o conversamente, astrazione). Una regione R' al livello $i+1$ raffina una regione R al livello i se R' è contenuta in R . Equivalentemente, R è un'astrazione di R' . In una gerarchia di partizione perfetta, ogni regione al livello i è partizionata esattamente dalle regioni al livello $i+1$ che la raffinano, senza sovrapposizioni e senza gap.

Formalmente, data una regione R in P_i , l'insieme di raffinamenti di R in P_{i+1} è: $\text{Refine}(R, P_{i+1}) = \{R' \in P_{i+1} : R' \subseteq R\}$. La proprietà di partizione richiede che l'unione di questo insieme sia esattamente R : $\bigcup \text{Refine}(R, P_{i+1}) = R$, e che gli elementi siano mutuamente disgiunti: $R'_1 \cap R'_2 = \emptyset$ per ogni coppia distinta $R'_1, R'_2 \in \text{Refine}(R, P_{i+1})$.

La relazione di raffinamento è transitiva: se R' raffina R e R'' raffina R' , allora R'' raffina R . Questo permette di saltare livelli intermedi nella gerarchia quando necessario. Possiamo chiedere direttamente quale insieme di regioni al livello fine n raffina una regione al livello grosso i , senza dover esplicitamente attraversare tutti i livelli intermedi.

La relazione inversa di astrazione mappa regioni fini a regioni grosse: $\text{Abstract}(R', P_i) = \{R \in P_i : R' \subseteq R\}$. In una gerarchia di partizione perfetta, questo insieme è sempre un singleton: ogni regione fine appartiene a esattamente una regione grossa al livello superiore. In gerarchie più generali (coperture invece di partizioni), una regione fine può sovrapporsi a multiple regioni grosse, rendendo l'astrazione ambigua.

Corrispondenze Cross-Scale e Alignment

Quando abbiamo due gerarchie di granularità indipendenti sullo stesso spazio (per esempio, divisioni amministrative e unità ecologiche), è importante stabilire corrispondenze o alignment tra esse. Una corrispondenza cross-scale mappa regioni di una gerarchia a regioni dell'altra, possibilmente a livelli di granularità diversi.

L'alignment può essere basato su criteri geometrici (sovrapposizione spaziale) o semantici (similitudine funzionale o tematica). L'alignment geometrico più semplice è quello basato sulla massima sovrapposizione: una regione R nella gerarchia A è allineata con la regione S nella gerarchia B se S è la regione in B con la maggiore area di intersezione con R .

Formalmente: $\text{Align}(R, H_B) = \arg \max_{\{S \in H_B\}} |R \cap S|$, dove H_B è l'unione di tutti i livelli della gerarchia B e $|\cdot|$ denota l'area o la misura. Questo può essere raffinato considerando non solo la sovrapposizione assoluta ma anche la sovrapposizione relativa: la frazione di R che è coperta da S e la frazione di S che è coperta da R .

Le corrispondenze cross-scale sono raramente perfect matches: è inusuale che i confini di regioni in due gerarchie indipendenti coincidano esattamente. Di conseguenza, l'alignment introduce un'ulteriore fonte di incertezza e imprecisione, che deve essere gestita formalmente. Tecniche di fuzzy matching assegnano gradi di allineamento invece di corrispondenze binarie, catturando il fatto che una regione amministrativa può corrispondere parzialmente a più regioni ecologiche.

Operatori di Zoom Concettuale

Gli operatori di zoom concettuale formalizzano il processo di cambiamento del livello di granularità nella rappresentazione. L'operatore di zoom-in prende una rappresentazione al livello i e produce una rappresentazione più dettagliata al livello $i+1$, rivelando la struttura interna delle regioni. L'operatore di zoom-out effettua l'operazione inversa, aggregando dettagli fini in regioni più grosse.

Lo zoom-in può essere deterministico o non-deterministico. In una gerarchia completamente specificata, lo zoom-in è deterministico: conosciamo esattamente come ogni regione al livello i è suddivisa al livello $i+1$. Tuttavia, quando la gerarchia è specificata solo parzialmente o quando abbiamo incertezza, lo zoom-in può essere non-deterministico: esistono molteplici raffinamenti possibili compatibili con l'informazione disponibile.

Formalmente, l'operatore di zoom-in può essere modellato come una funzione $\text{ZoomIn}: 2^{\{P_i\}} \rightarrow 2^{\{2^{\{P_{i+1}\}}\}}$ che prende un insieme di regioni al livello i e restituisce un insieme di insiemi di regioni al livello $i+1$ (ciascuno rappresentando un possibile raffinamento). Nel caso deterministico, l'insieme esterno è singleton. Nel caso non-deterministico, contiene molteplici alternative.

Lo zoom-out è tipicamente deterministico anche in presenza di incertezza: data una collezione di regioni al livello $i+1$, l'operatore di zoom-out le raggruppa nelle corrispondenti regioni al livello i secondo la struttura gerarchica. $\text{ZoomOut}: 2^{\{P_{i+1}\}} \rightarrow 2^{\{P_i\}}$ mappa semplicemente ogni regione fine alla regione grossa che la contiene, e poi rimuove duplicati.

Propagazione di Informazione tra Livelli

Un aspetto cruciale del reasoning multi-granularità è la propagazione di informazione tra livelli. Se conosciamo una proprietà a un certo livello, cosa possiamo inferire sugli altri livelli?

La propagazione ascendente (bottom-up) aggrega informazione da livelli fini a livelli grossi. Se tutte le sotto-regioni di R al livello fine hanno una certa proprietà P , allora possiamo inferire che R ha la proprietà P al livello grosso (per proprietà universali). Se almeno una sotto-regione di R ha la proprietà P , allora R ha possibilmente la proprietà P (per proprietà esistenziali). Per proprietà quantitative (come popolazione o temperatura media), la propagazione ascendente coinvolge operazioni di aggregazione: somma, media, massimo, minimo, a seconda della natura della proprietà.

La propagazione discendente (top-down) distribuisce informazione da livelli grossi a livelli fini. Se una regione R al livello grosso ha una proprietà P , cosa possiamo dire sulle sue sotto-regioni al livello fine? Nel caso più semplice, se P è ereditabile, allora tutte le sotto-regioni hanno P . Ma spesso l'informazione al livello grosso è compatibile con molteplici distribuzioni al livello fine. Per esempio, se sappiamo che la popolazione totale di una regione è centomila abitanti, questo è compatibile con molte distribuzioni possibili della popolazione tra le sotto-regioni.

Il problema del downscaling probabilistico considera la propagazione discendente sotto incertezza. Data una distribuzione di probabilità su proprietà al livello grosso, quale distribuzione di probabilità possiamo inferire al livello fine? Tecniche di maximum entropy o metodi bayesiani possono essere utilizzati per scegliere la distribuzione fine "più probabile" o "meno committente" compatibile con i vincoli grossi.

Reasoning Qualitativo Multi-Granularità

Il ragionamento qualitativo spaziale deve affrontare sfide specifiche quando opera su rappresentazioni multi-granularità. Le relazioni topologiche, mereologiche e direzionali possono essere definite a ogni livello di granularità, ma la loro semantica e le loro proprietà di inferenza possono dipendere dal livello.

Topologia Relative a Granularità

Una relazione topologica tra regioni può essere vera a un livello di granularità ma falsa a un altro. Consideriamo due regioni A e B che sono disconnesse (DC) a un livello di granularità grossa, perché le loro rappresentazioni grosse non si toccano. Tuttavia, a un livello di granularità fine, potrebbe emergere che i confini dettagliati di A e B effettivamente si toccano in un punto, rendendo le regioni esternalmente connesse (EC) invece di disconnesse.

Questo fenomeno è chiamato granularity-dependent topology. Per formalizzarlo, dobbiamo parametrizzare le relazioni topologiche rispetto al livello di granularità. Scriviamo $DC_i(A, B)$ per indicare che A e B sono disconnesse alla granularità del livello i . La consistenza gerarchica richiede che certe transizioni topologiche siano permesse tra livelli, mentre altre siano proibite.

Per esempio, se $DC_i(A, B)$ è vero, allora a un livello più fine $i+1$, le relazioni possibili sono $DC_{i+1}(A, B)$ (ancora disconnesse), $EC_{i+1}(A, B)$ (ora esternalmente connesse perché i dettagli rivelano un punto di contatto), o $PO_{i+1}(A, B)$ (overlap parziale se il confine grossolano era un'approssimazione e in realtà c'è una piccola zona di sovrapposizione). Non è permesso che $DC_i(A, B)$ si trasformi in $TPP_{i+1}(A, B)$ (A parte tangenziale propria di B), perché questo richiederebbe che A fosse dentro B , contraddicendo il fatto che erano disconnesse al livello grosso.

Vincoli formali sulla monotonia delle relazioni topologiche tra livelli garantiscono la coerenza. Un framework per questo è il Spatial Granularity RCC, che estende il Region Connection Calculus con indici di granularità e assiomi che specificano quali transizioni di relazioni RCC sono permesse durante il raffinamento o l'astrazione.

Mereologia a Diverse Scale

Le relazioni mereologiche (parte-tutto) sono naturalmente legate alla granularità. A un livello grosso, potremmo dire che la regione A è parte della regione B, ma a un livello fine, potremmo scoprire che A in realtà consiste di molteplici frammenti disconnessi, alcuni dei quali sono dentro B e altri fuori da B.

La nozione di parte può essere relativizzata alla scala: A è parte di B alla scala s se, alla rappresentazione di dettaglio s , l'intera estensione di A è contenuta nell'estensione di B. Questa definizione può produrre risultati diversi a scale diverse se i confini di A o B sono approssimazioni dipendenti dalla scala.

Una questione sottile riguarda la transitività della relazione di parte attraverso scale. Se A è parte di B alla scala s_1 , e B è parte di C alla scala s_2 (con s_1 più fine di s_2), possiamo concludere che A è parte di C? In generale, sì, se assumiamo consistenza gerarchica: se $A \subseteq B$ alla scala fine, allora $A \subseteq B$ anche a qualsiasi scala grossa (perché l'astrazione preserva la relazione di inclusione). Ma se le scale s_1 e s_2 non sono gerarchicamente ordinate, la transitività non è garantita.

Direzioni Qualitative e Granularità

Le relazioni direzionali qualitative (nord, sud, est, ovest) possono anch'esse dipendere dalla granularità. A un livello molto grosso, una città B potrebbe essere a nord di una città A. A un livello più fine, potremmo vedere che il centro di B è leggermente a nord-est del centro di A. A un livello ancora più fine, alcune parti di B possono essere a nord di A mentre altre parti sono a est o a nord-est.

Questo solleva la questione di come definire relazioni direzionali per regioni estese a diverse scale. Un approccio è quello di utilizzare i centroidi o i punti rappresentativi delle regioni: la relazione direzionale tra A e B è determinata dalla direzione dal centroide di A al centroide di B. Ma la posizione del centroide può cambiare con il livello di dettaglio se la forma della regione viene raffinata.

Un approccio alternativo è quello di definire relazioni direzionali qualitative in termini di intervalli di direzioni: B è "approssimativamente a nord" di A se la maggior parte della massa di B giace in un settore angolare centrato sulla direzione nord rispetto ad A. La larghezza del settore può essere correlata al livello di granularità: a granularità grossa, usiamo settori ampi (per esempio, nord copre da 45° a 135°), mentre a granularità fine, usiamo settori stretti (nord copre da 80° a 100°).

Consistency Checking Cross-Scale

Un problema fondamentale nel reasoning multi-granularità è il consistency checking: dato un insieme di asserzioni spaziali a diversi livelli di granularità, esiste una configurazione spaziale globale compatibile con tutte le asserzioni? Questo problema è significativamente più complesso del consistency checking a singola granularità, perché dobbiamo verificare non solo la consistenza interna a ogni livello, ma anche la consistenza tra livelli.

Un approccio è quello di propagare vincoli tra livelli fino a raggiungere un punto fisso. Iniziamo con gli insiemi di configurazioni possibili a ogni livello, derivati direttamente dalle asserzioni a quel livello. Poi iteriamo: per ogni livello i , usiamo le configurazioni possibili al livello $i-1$ e $i+1$ per restringere le configurazioni possibili al livello i , applicando regole di compatibilità cross-scale. Continuiamo finché non si verificano ulteriori cambiamenti. Se a qualche livello l'insieme di configurazioni possibili diventa vuoto, il sistema è inconsistente. Altrimenti, qualsiasi configurazione sopravvissuta a tutti i livelli è globalmente consistente.

La complessità di questo algoritmo dipende dalla struttura della gerarchia e dalla natura dei vincoli. Nel caso peggiore, può essere esponenziale nel numero di regioni e livelli. Tuttavia, euristiche basate sulla struttura gerarchica e su ordinamenti intelligenti delle propagazioni possono rendere il problema trattabile in molti casi pratici.

Rappresentazioni Multi-Risoluzione: Strutture Dati e Algoritmi

Dal punto di vista computazionale, la gestione efficiente di dati spaziali multi-granularità richiede strutture dati specializzate che supportano query e aggiornamenti a diversi livelli di dettaglio.

Quadrees e Octrees

Le quadrees sono strutture ad albero per la rappresentazione gerarchica di dati spaziali bidimensionali. Il concetto è semplice: lo spazio è ricorsivamente suddiviso in quadranti. Alla radice dell'albero, abbiamo l'intero spazio. Ogni nodo interno rappresenta una regione quadrata, che è divisa in quattro quadranti figli di dimensione uguale. Le foglie dell'albero rappresentano regioni che non sono ulteriormente suddivise, o perché hanno raggiunto la risoluzione massima, o perché sono omogenee secondo qualche criterio.

Formalmente, un nodo di una quadtree a profondità d rappresenta una regione di dimensione $2^{-d} \times 2^{-d}$ (normalizzando lo spazio totale a 1×1). I quattro figli di questo nodo rappresentano i quattro quadranti di questa regione: nord-ovest, nord-est, sud-ovest, sud-est. La profondità dell'albero determina la risoluzione massima: una quadtree di profondità D ha risoluzione minima 2^{-D} .

Le quadrees sono particolarmente efficienti per dati che esibiscono coerenza spaziale: regioni larghe con proprietà uniforme possono essere rappresentate da nodi interni di alta granularità senza espandere i loro figli, risparmiando memoria. Solo le regioni con dettagli fini o variazioni richiedono espansione fino ai livelli più profondi.

Gli octrees estendono questo concetto allo spazio tridimensionale: ogni nodo interno ha otto figli corrispondenti agli otto ottanti del cubo che rappresenta. Sono utilizzati in grafica computazionale per la rappresentazione di oggetti tridimensionali, in simulazioni scientifiche per griglia adattive, e in robotica per occupancy maps tridimensionali.

Piramidi di Immagini e Wavelets

Le piramidi di immagini sono strutture multi-risoluzione ampiamente utilizzate in visione computazionale e elaborazione di immagini. Una piramide di immagini consiste di una sequenza di immagini dello stesso soggetto a risoluzioni progressivamente più grosse. Il livello zero è l'immagine originale a piena risoluzione. Ogni livello successivo è ottenuto sottocampionando il livello precedente, tipicamente riducendo la risoluzione di un fattore due in ciascuna dimensione.

Il processo di sottocampionamento non è semplicemente la selezione di ogni secondo pixel, perché questo causerebbe aliasing. Invece, ogni livello è ottenuto applicando un filtro di smoothing (tipicamente un filtro Gaussiano) seguito da sottocampionamento. Questo produce la Gaussian pyramid. Varianti come la Laplacian pyramid memorizzano la differenza tra livelli consecutivi, permettendo una ricostruzione perfetta dell'immagine originale.

Le trasformate wavelet forniscono un framework matematicamente rigoroso per l'analisi multi-risoluzione. Una trasformata wavelet decompone un segnale (o un'immagine) in componenti a diverse scale e posizioni spaziali. I coefficienti wavelet a scale grosse catturano le caratteristiche di bassa frequenza (tendenze generali), mentre i coefficienti a scale fini catturano dettagli ad alta frequenza (bordi, texture).

Una proprietà importante delle wavelets è la loro località sia nello spazio che nella frequenza: ogni coefficiente wavelet è associato a una specifica regione spaziale e a una specifica banda di frequenza. Questo le rende ideali per l'analisi multi-scala di dati spaziali: possiamo studiare fenomeni a larga scala osservando coefficienti wavelet di scale grosse, e studiare dettagli locali osservando coefficienti di scale fini in regioni specifiche.

R-Trees Multi-Livello

Gli R-trees sono strutture dati per l'indicizzazione di oggetti spaziali, ampiamente utilizzati in database spaziali. Un R-tree è un albero bilanciato dove ogni nodo rappresenta un minimum bounding rectangle (MBR) che contiene tutti gli oggetti nei suoi sottoalberi. Le foglie dell'albero puntano agli oggetti spaziali effettivi, mentre i nodi interni servono come indice gerarchico.

Per supportare query multi-granularità, gli R-trees possono essere estesi con informazione aggregata a ogni nodo. Per esempio, oltre al MBR, ogni nodo può memorizzare statistiche sugli oggetti contenuti: numero totale, densità media, distribuzione di attributi tematici. Questo permette di rispondere a query aggregate a diversi livelli di granularità senza dover scendere alle foglie.

Un'altra estensione è quella di mantenere multiple rappresentazioni di ogni oggetto a diverse risoluzioni, e di indicizzarle in un R-tree multi-livello. Ogni oggetto geometrico complesso è approssimato da rappresentazioni semplificate a diverse scale: una rappresentazione grossolana con pochi vertici, rappresentazioni intermedie con dettaglio crescente, fino alla rappresentazione completa. Le query possono prima controllare le rappresentazioni grossolane per filtrare rapidamente oggetti irrilevanti, e poi raffinare i risultati usando rappresentazioni più dettagliate solo per i candidati rimanenti.

Level-of-Detail (LOD) Management

Nella grafica computazionale e nei sistemi di informazione geografica tridimensionali, la gestione del Level-of-Detail è essenziale per performance interattive. Un modello tridimensionale complesso (per esempio, il terreno di una vasta area o un modello di città) non può essere renderizzato completamente a massimo dettaglio in tempo reale, specialmente se alcune parti sono lontane dalla camera o periferiche nel campo visivo.

La soluzione è quella di mantenere rappresentazioni a diversi livelli di dettaglio e di selezionare dinamicamente il LOD appropriato per ogni parte della scena basandosi su criteri come distanza dalla camera, dimensione proiettata sullo schermo, importanza semantica, o budget computazionale. Regioni vicine alla camera sono renderizzate con LOD alto (molti poligoni, texture ad alta risoluzione), mentre regioni lontane usano LOD basso (geometria semplificata, texture di bassa risoluzione).

Gli algoritmi di LOD selection devono gestire transizioni fluide tra livelli di dettaglio per evitare artefatti visivi come "popping" (apparizione improvvisa di dettagli). Tecniche di geomorphing interpolano gradualmente tra geometrie a diversi LOD, mentre tecniche di blending mescolano texture a diverse risoluzioni in modo smooth.

Il framework formale sottostante è una gerarchia di semplificazioni: ogni modello al LOD $i+1$ è una versione semplificata del modello al LOD i , ottenuta attraverso operazioni come decimazione di mesh (rimozione di vertici e facce secondo criteri che minimizzano l'errore geometrico), o texture filtering (riduzione di risoluzione di texture). La struttura gerarchica permette di navigare efficientemente nello spazio dei LOD in risposta a cambiamenti dinamici nella scena.

Logiche Cinematiche e Dinamiche

Introduzione alla Rappresentazione del Movimento Spaziale

La rappresentazione formale del movimento e delle dinamiche spaziali costituisce una delle sfide più complesse nella rappresentazione della conoscenza spaziale. Mentre le logiche topologiche e mereologiche si concentrano sulla struttura statica dello spazio, le logiche cinematiche e dinamiche introducono la dimensione temporale, permettendo di ragionare su come gli oggetti si muovono, interagiscono e cambiano le loro relazioni spaziali nel tempo. Questo ambito di ricerca è fondamentale per applicazioni che vanno dalla robotica mobile alla comprensione di scene video, dalla simulazione fisica alla pianificazione di traiettorie, fino all'analisi di dati geospaziali temporali.

Il movimento spaziale può essere considerato a diversi livelli di astrazione. Al livello più concreto, abbiamo traiettorie parametriche definite da funzioni continue dello spazio rispetto al tempo. A un livello più astratto, possiamo descrivere il movimento in termini qualitativi, catturando proprietà essenziali come la direzione relativa del movimento, la convergenza o divergenza di oggetti mobili, il cambiamento delle relazioni topologiche durante il movimento.

Questa astrazione qualitativa è particolarmente importante quando i dettagli metrici precisi non sono disponibili o non sono rilevanti per il ragionamento richiesto.

Le logiche cinematiche si occupano della descrizione geometrica del movimento senza considerare le forze che lo causano, mentre le logiche dinamiche incorporano anche le cause del movimento e le leggi fisiche che lo governano. In questo trattato ci concentreremo principalmente sugli aspetti cinematici e qualitativi, che hanno ricevuto maggiore attenzione nella ricerca sulla rappresentazione della conoscenza spaziale.

Qualitative Trajectory Calculus (QTC)

Il Qualitative Trajectory Calculus rappresenta uno dei contributi più significativi alla rappresentazione qualitativa del movimento. Sviluppato principalmente da Van de Weghe e colleghi nei primi anni del duemila, il QTC fornisce un framework sistematico per descrivere e ragionare sul movimento relativo tra oggetti mobili. L'idea centrale è quella di astrarre dalle coordinate metriche precise e concentrarsi sulle caratteristiche qualitative del movimento che sono rilevanti per il ragionamento umano e per molte applicazioni pratiche.

QTC Base (QTC_B): Movimento in una Dimensione

La versione più semplice del QTC, denominata QTC_B, considera il movimento di due punti k e l lungo una linea retta. Per ogni istante temporale, il calcolo cattura tre aspetti fondamentali del movimento relativo:

Il primo parametro caratterizza il movimento di k rispetto al punto di riferimento l . Questo parametro può assumere tre valori: meno se k si sta muovendo verso l , zero se k è stazionario rispetto a l , oppure più se k si sta allontanando da l . Formalmente, se indichiamo con $d_{kl}(t)$ la distanza tra k e l al tempo t , allora questo primo parametro è determinato dal segno della derivata temporale di $d_{kl}(t)$ dal punto di vista di k .

Il secondo parametro è simmetrico e caratterizza il movimento di l rispetto a k . Anche questo può essere meno, zero o più, a seconda che l si stia avvicinando, sia stazionario o si stia allontanando rispetto a k .

Il terzo parametro cattura la velocità relativa tra i due oggetti. Più precisamente, indica se la distanza tra k e l sta diminuendo (meno), rimanendo costante (zero), o aumentando (più). Questo parametro è determinato dal segno della derivata seconda della funzione distanza, o equivalentemente, dalla somma vettoriale delle velocità relative.

La notazione standard per una configurazione QTC_B è una tripla di simboli. Per esempio, la configurazione (meno, più, meno) indica che k si sta muovendo verso l , l si sta allontanando da k , e complessivamente la distanza tra i due sta diminuendo. Questo significa che k si sta muovendo più velocemente verso l di quanto l si stia allontanando da k .

Non tutte le combinazioni dei tre parametri sono concettualmente coerenti. Delle ventisette combinazioni teoricamente possibili, solo alcune rappresentano configurazioni fisicamente realizzabili. Per esempio, se entrambi gli oggetti sono stazionari (zero, zero), allora la

distanza non può cambiare, quindi il terzo parametro deve essere necessariamente zero. Questo porta a un insieme di tredici configurazioni valide per QTC_B.

QTC Doppio Croce (QTC_C): Movimento sul Piano

L'estensione al movimento bidimensionale, chiamata QTC_C, introduce una complessità significativamente maggiore ma anche una capacità espressiva molto più ricca. Nel caso bidimensionale, dobbiamo considerare non solo se gli oggetti si avvicinano o si allontanano, ma anche la direzione relativa del loro movimento.

QTC_C utilizza un sistema di riferimento intrinseco definito dalla linea che connette i due oggetti. Immaginiamo di essere posizionati in *k* e di guardare verso *l*. Possiamo quindi definire un asse longitudinale lungo la linea *k-l* e un asse trasversale perpendicolare ad esso. Il movimento di ciascun oggetto può essere decomposto lungo questi due assi, generando una rappresentazione quadridimensionale del movimento relativo.

I primi due parametri descrivono il movimento longitudinale, esattamente come in QTC_B: il movimento di *k* verso o lontano da *l*, e il movimento di *l* verso o lontano da *k*. I parametri terzo e quarto descrivono il movimento trasversale: il movimento laterale di *k* rispetto alla linea *k-l*, e il movimento laterale di *l* rispetto alla stessa linea. Per il movimento trasversale, utilizziamo le convenzioni meno per il movimento verso sinistra (dal punto di vista di un osservatore in *k* che guarda verso *l*), zero per nessun movimento laterale, e più per il movimento verso destra.

Questa rappresentazione è particolarmente potente per catturare pattern di movimento come sorpasso, attraversamento, inseguimento, evasione e molti altri comportamenti spazialmente rilevanti. Per esempio, una configurazione (meno, più, zero, meno) indica che *k* si sta avvicinando a *l* mentre *l* si sta allontanando da *k*, *k* non ha movimento laterale ma *l* si sta muovendo lateralmente verso sinistra, suggerendo un possibile pattern di evasione dove *l* cerca di evitare *k* muovendosi lateralmente mentre si allontana.

QTC con Distanza (QTC_D): Incorporazione di Informazione Metrica Qualitativa

Una limitazione del QTC puro è che non considera esplicitamente la distanza tra gli oggetti, solo la sua variazione. QTC_D aggiunge un ulteriore parametro che qualifica la distanza corrente come vicina, media o lontana rispetto a soglie contestualmente determinate. Questo permette di distinguere, per esempio, tra un avvicinamento che avviene quando gli oggetti sono già vicini (potenzialmente rilevante per la previsione di collisioni) e un avvicinamento che avviene quando sono ancora lontani.

La definizione delle soglie di distanza può essere assoluta (basata su misure metriche fisse) o relativa (basata sulla dimensione degli oggetti, sulla velocità del movimento, o su altri fattori contestuali). In molte applicazioni, la scelta delle soglie è cruciale per ottenere un livello di granularità appropriato nella descrizione del movimento.

Relazioni di Transizione in QTC

Un aspetto fondamentale del QTC è la sua capacità di modellare le transizioni tra configurazioni. Non tutte le transizioni sono possibili: esistono vincoli sulle sequenze di configurazioni che possono verificarsi nel tempo. Questi vincoli possono essere rappresentati mediante un grafo di transizione concettuale, dove i nodi rappresentano le configurazioni QTC e gli archi rappresentano le transizioni possibili.

Per esempio, in QTC_B, la transizione dalla configurazione (meno, più, meno) può portare direttamente solo a un sottoinsieme limitato di configurazioni. Se k continua ad avvicinarsi più velocemente di quanto l si allontani, potremmo raggiungere la configurazione dove i due oggetti si incontrano. Se invece le velocità cambiano, potremmo transitare a configurazioni dove la distanza inizia ad aumentare.

L'analisi formale di questi grafi di transizione rivela proprietà importanti. Alcune configurazioni sono stabili, nel senso che possono persistere per intervalli temporali estesi senza necessariamente transitare ad altre configurazioni. Altre configurazioni sono istantanee, rappresentando momenti di transizione che non possono durare. Per esempio, la configurazione dove entrambi gli oggetti sono esattamente stazionari per un istante ma stanno per iniziare a muoversi rappresenta un punto di transizione.

Composizione e Reasoning in QTC

Il QTC supporta operazioni di composizione che permettono di inferire relazioni di movimento tra oggetti che non sono direttamente osservati insieme. Se conosciamo la relazione QTC tra gli oggetti A e B, e la relazione QTC tra B e C, possiamo in alcuni casi derivare vincoli sulla relazione tra A e C. Tuttavia, la composizione in QTC è più complessa che nei calcoli puramente topologici o mereologici, perché il movimento è intrinsecamente relativo al sistema di riferimento scelto.

Formalmente, data una configurazione q_1 tra A e B e una configurazione q_2 tra B e C, la composizione $q_1 \circ q_2$ produce un insieme di configurazioni possibili tra A e C. Questo insieme può essere molto ampio se le configurazioni iniziali non vincolano fortemente il movimento relativo. Tecniche di propagazione di vincoli possono essere utilizzate per raffinare progressivamente gli insiemi di configurazioni possibili, dato un insieme di osservazioni parziali.

Il problema del consistency checking in QTC chiede se un insieme di vincoli sul movimento relativo tra coppie di oggetti ammette almeno una configurazione globale coerente. Questo problema è computazionalmente complesso, ma esistono algoritmi efficienti per sottoclassi ristrette del problema, basati su decomposizione gerarchica o su rappresentazioni a grafo delle dipendenze tra oggetti.

Formalismi per il Reasoning su Movimento Relativo tra Oggetti

Oltre al QTC, esistono diversi altri formalismi che catturano aspetti specifici del movimento relativo. Questi approcci complementari offrono diversi trade-off tra espressività, efficienza computazionale e naturalezza cognitiva.

Calculus of Moving Objects (CMO)

Il Calculus of Moving Objects si concentra su come le relazioni topologiche tradizionali (come quelle del RCC-8) cambiano durante il movimento. L'idea centrale è quella di annotare le relazioni topologiche con informazione sul loro cambiamento temporale. Per esempio, invece di dire semplicemente che due regioni A e B sono esternamente connesse (EC), possiamo dire che sono EC e questa relazione sta diventando più stretta, suggerendo un movimento di avvicinamento.

Formalmente, il CMO introduce operatori temporali che modificano le relazioni RCC-8 di base. Un operatore di tendenza indica la direzione del cambiamento: una relazione può essere stabile (la configurazione topologica non sta cambiando), rinforzante (gli oggetti stanno diventando più strettamente connessi topologicamente), o indebolente (gli oggetti stanno diventando meno connessi). Per esempio, se due regioni sono esternamente connesse e il loro movimento le sta portando a sovrapporsi, abbiamo una relazione EC rinforzante che probabilmente transiterà a una relazione di overlap parziale.

Il CMO definisce anche vincoli sulla dinamica delle transizioni topologiche. Non tutte le sequenze di relazioni RCC-8 sono realizzabili attraverso movimento continuo nello spazio. Per esempio, due regioni disgiunte non possono diventare improvvisamente identiche senza passare attraverso configurazioni intermedie di connessione e overlap. Questi vincoli di continuità topologica sono fondamentali per il reasoning su scenari dinamici.

Direction-of-Motion Calculi

I calcoli basati sulla direzione del movimento si concentrano sull'orientamento del vettore velocità degli oggetti. Questi formalismi sono particolarmente utili quando la direzione del movimento è più saliente della distanza o della topologia.

Un approccio comune è quello di discretizzare lo spazio delle direzioni possibili in un numero finito di settori. Per esempio, possiamo dividere i trecentosessanta gradi attorno a un oggetto in otto settori principali: nord, nord-est, est, sud-est, sud, sud-ovest, ovest, nord-ovest. La direzione del movimento di un oggetto è quindi classificata in uno di questi settori.

Le relazioni direzionali del movimento possono essere combinate con relazioni topologiche per ottenere descrizioni più ricche. Per esempio, possiamo dire che l'oggetto A si sta muovendo verso nord mentre è a nord-ovest dell'oggetto B, fornendo sia informazione posizionale che cinematica.

Un'estensione più sofisticata considera la direzione del movimento relativo tra due oggetti. Dato un sistema di riferimento centrato sull'oggetto A, possiamo classificare la direzione in cui B si sta muovendo rispetto ad A. Questo è diverso dalla direzione assoluta del movimento di B, perché tiene conto anche del movimento di A stesso. Se entrambi gli oggetti si muovono verso nord ma A è più veloce, nel sistema di riferimento di A sembrerà che B si stia muovendo verso sud.

Velocity-Based Qualitative Reasoning

Alcuni formalismi si concentrano sulla velocità del movimento piuttosto che sulla direzione. Il reasoning qualitativo sulla velocità divide i valori possibili di velocità in intervalli qualitativi: fermo, lento, medio, veloce, molto veloce. Le transizioni tra questi stati e le loro combinazioni per oggetti multipli generano uno spazio di configurazioni cinematiche qualitative.

Un aspetto interessante è il confronto relativo di velocità. Possiamo dire che l'oggetto A si sta muovendo più velocemente dell'oggetto B senza specificare i valori esatti di velocità. Questo tipo di reasoning ordinale è robusto rispetto all'incertezza nelle misurazioni e cattura informazione spesso sufficiente per molti compiti di ragionamento.

La composizione di relazioni di velocità segue regole algebriche intuitive. Se A è più veloce di B e B è più veloce di C, allora possiamo inferire con certezza che A è più veloce di C. Tuttavia, se A è più veloce di B e C è più veloce di D, non possiamo in generale inferire nulla sulla relazione tra B e C senza ulteriore informazione.

Accelerazione e Cambiamento di Velocità

Estensioni più avanzate dei formalismi cinematici considerano anche l'accelerazione, cioè il tasso di cambiamento della velocità. In termini qualitativi, possiamo distinguere tra oggetti che si muovono a velocità costante, oggetti che stanno accelerando (aumentando la loro velocità), e oggetti che stanno decelerando (diminuendo la loro velocità).

L'accelerazione è particolarmente importante per la previsione del comportamento futuro. Un oggetto che si sta muovendo velocemente ma sta decelerando fortemente potrebbe fermarsi presto, mentre un oggetto che si muove lentamente ma sta accelerando potrebbe diventare rapidamente veloce. Questi pattern di cambiamento sono cruciali per compiti come la pianificazione di traiettorie e la previsione di collisioni.

In una rappresentazione qualitativa dell'accelerazione, possiamo classificare il cambiamento di velocità come: accelerazione positiva forte, accelerazione positiva debole, velocità costante, accelerazione negativa debole, accelerazione negativa forte. Questa classificazione può essere ulteriormente raffinata o semplificata a seconda del livello di granularità richiesto dall'applicazione.

Logiche per la Previsione di Collisioni

La previsione di collisioni tra oggetti in movimento è un problema fondamentale con applicazioni che vanno dalla robotica mobile alla simulazione fisica, dai videogiochi alla pianificazione del traffico. Le logiche formali per la previsione di collisioni devono catturare non solo la geometria e la cinematica corrente, ma anche l'evoluzione futura del sistema.

Collision Zones e Time to Collision

Un concetto centrale nella previsione di collisioni è quello di zona di collisione. Data la traiettoria prevista di un oggetto, la zona di collisione è la regione dello spazio che l'oggetto attraverserà in un determinato intervallo di tempo futuro. Se le zone di collisione di due oggetti si intersecano, allora esiste la possibilità di una collisione.

Formalmente, se un oggetto A ha posizione $p_A(t)$ al tempo t e si muove con velocità v_A , la sua posizione al tempo futuro $t + \Delta t$ sarà approssimativamente $p_A(t + \Delta t) \approx p_A(t) + v_A \cdot \Delta t$, assumendo velocità costante. La zona di collisione per l'intervallo $[t, t + \Delta t]$ è l'unione di tutte le posizioni occupate dall'oggetto durante questo intervallo, tenendo conto della sua forma geometrica.

Il time to collision (TTC) tra due oggetti è il tempo minimo prima che si verifichi una collisione, assumendo che entrambi gli oggetti continuino a muoversi secondo le loro velocità correnti. Se gli oggetti non sono su una traiettoria di collisione, il TTC è infinito. In una rappresentazione qualitativa, possiamo classificare il TTC in categorie come: collisione imminente (TTC molto breve), collisione possibile nel breve termine, collisione possibile nel medio termine, nessuna collisione prevista.

Calcoli di Collision Threat

I calcoli di minaccia di collisione combinano informazione su distanza, velocità relativa e direzione del movimento per valutare il rischio di collisione. Un approccio comune è quello di definire una funzione di minaccia che aumenta quando gli oggetti sono vicini, si muovono velocemente uno verso l'altro, e le loro traiettorie sono convergenti.

Una formula semplice per una misura qualitativa di minaccia potrebbe essere: $\text{Threat} \propto \text{RelativeVelocity} / \text{Distance}$, dove RelativeVelocity è la componente della velocità relativa nella direzione che connette i due oggetti, e Distance è la distanza corrente tra essi. Questa misura diventa grande quando gli oggetti sono vicini e si muovono velocemente uno verso l'altro.

In un framework qualitativo, possiamo discretizzare questa misura di minaccia in livelli: nessuna minaccia, minaccia bassa, minaccia media, minaccia alta, minaccia critica. Soglie appropriate per questi livelli dipendono dal contesto applicativo e dalla capacità di reazione del sistema. Per un robot mobile che può cambiare direzione rapidamente, la minaccia diventa critica solo a distanze molto brevi, mentre per un veicolo pesante con inerzia elevata, la minaccia diventa critica a distanze maggiori.

Reasoning su Evitamento di Collisioni

Oltre a prevedere collisioni, le logiche cinematiche possono ragionare su strategie di evitamento. Data una configurazione cinematica corrente e un insieme di azioni possibili (cambiamenti di velocità o direzione), quali azioni permettono di evitare una collisione prevista?

Un approccio formale considera lo spazio delle traiettorie possibili come un albero di decisione. Ogni nodo rappresenta una configurazione cinematica (posizioni e velocità degli oggetti), e ogni ramo rappresenta una scelta di azione. Una strategia di evitamento è un percorso attraverso questo albero che non passa attraverso configurazioni di collisione.

La complessità computazionale del planning di evitamento cresce rapidamente con il numero di oggetti e la complessità delle loro forme geometriche. Tuttavia, astrazioni qualitative possono rendere il problema più trattabile. Per esempio, invece di considerare

tutte le possibili velocità continue, possiamo considerare solo un numero finito di manovre qualitative: accelera, decelera, gira a sinistra, gira a destra, mantieni la rotta.

Logiche Temporal per Proprietà di Sicurezza

Le logiche temporali, in particolare Linear Temporal Logic (LTL) e Computation Tree Logic (CTL), possono essere utilizzate per specificare e verificare proprietà di sicurezza in sistemi con oggetti in movimento. Una proprietà di sicurezza tipica è: "non si verifica mai una collisione", formalizzata in LTL come $G(\neg \text{collision})$, dove G è l'operatore "sempre" (globally) e collision è una proposizione atomica che diventa vera quando si verifica una collisione.

Proprietà più sofisticate possono esprimere vincoli condizionali: "se un oggetto entra in una zona pericolosa, allora eventualmente deve uscirne senza collisioni", formalizzata come $G(\text{enterDangerZone} \rightarrow F(\text{exitDangerZone} \wedge \neg \text{collision}))$, dove F è l'operatore "eventualmente" (finally).

Il model checking di queste proprietà su modelli cinematici può essere utilizzato per verificare automaticamente se un sistema di controllo soddisfa specifiche di sicurezza. Tuttavia, la presenza di tempo continuo e spazio continuo rende il model checking esatto spesso intrattabile. Tecniche di astrazione e discretizzazione sono necessarie per rendere la verifica praticabile, al costo di possibili risultati conservativi (falsi positivi nella rilevazione di violazioni di sicurezza).

Rappresentazione di Traiettorie Complesse

Le traiettorie reali degli oggetti sono spesso molto più complesse dei semplici movimenti rettilinei a velocità costante considerati nei formalismi qualitativi di base. Curve, accelerazioni, movimenti episodici con interruzioni, e pattern di movimento ripetitivi richiedono rappresentazioni più ricche.

Segmentazione Qualitativa di Traiettorie

Un approccio comune è quello di segmentare una traiettoria complessa in una sequenza di segmenti qualitativamente omogenei. Ogni segmento corrisponde a una fase del movimento dove le caratteristiche qualitative rimangono costanti. I punti di transizione tra segmenti sono eventi significativi: cambio di direzione, inizio o fine dell'accelerazione, raggiungimento di una regione spaziale particolare.

Formalmente, data una traiettoria continua $T: [0, T_{\max}] \rightarrow \text{Space}$, una segmentazione è una sequenza di intervalli temporali $[t_0, t_1], [t_1, t_2], \dots, [t_{n-1}, t_n]$ con $t_0 = 0$ e $t_n = T_{\max}$, tale che ciascun segmento $T|_{[t_i, t_{i+1}]}$ soddisfa un criterio di omogeneità qualitativa. La scelta del criterio di omogeneità dipende dalle caratteristiche del movimento che vogliamo catturare: direzione costante, velocità approssimativamente costante, accelerazione nulla, permanenza in una regione topologica.

La segmentazione automatica di traiettorie da dati grezzi è un problema non triviale, perché richiede di identificare i punti di cambiamento significativo in presenza di rumore e incertezza. Tecniche di smoothing e filtri come il filtro di Kalman possono essere utilizzati per

ridurre il rumore prima della segmentazione. Algoritmi di change-point detection possono identificare automaticamente gli istanti in cui le proprietà statistiche della traiettoria cambiano significativamente.

Pattern di Movimento Ricorrenti

Molti fenomeni naturali e artificiali esibiscono pattern di movimento ricorrenti o ciclici. Veicoli che percorrono regolarmente la stessa rotta, animali che migrano stagionalmente seguendo percorsi simili, satelliti in orbite periodiche, sono tutti esempi di movimento con struttura ripetitiva. La rappresentazione formale di questi pattern permette di comprimere l'informazione e di fare previsioni basate sulla periodicità osservata.

Un pattern di movimento può essere rappresentato come una sequenza prototipica di configurazioni qualitative che si ripete con variazioni minori. Per esempio, il pattern di un autobus urbano può essere rappresentato come una sequenza: fermo alla fermata A, in movimento verso la fermata B, fermo alla fermata B, in movimento verso la fermata C, e così via, con il pattern che si ripete ciclicamente lungo il percorso.

La matching di traiettorie osservate contro pattern noti è un problema di riconoscimento di pattern. Tecniche di dynamic time warping permettono di confrontare sequenze di diversa lunghezza, identificando allineamenti ottimali che massimizzano la similarità tra la traiettoria osservata e il pattern di riferimento. Questo è robusto rispetto a variazioni nella velocità di esecuzione del pattern (un autobus può impiegare tempi diversi a seconda del traffico) pur mantenendo la struttura qualitativa del movimento.

Rappresentazione di Incertezza nelle Traiettorie

Le traiettorie future sono intrinsecamente incerte. Anche se conosciamo perfettamente la posizione e la velocità correnti di un oggetto, non possiamo prevedere con certezza assoluta dove sarà in futuro, perché l'oggetto potrebbe cambiare direzione o velocità in risposta a eventi esterni o decisioni interne.

Una rappresentazione dell'incertezza nelle traiettorie è il trajectory funnel o cono di incertezza. Invece di una singola curva nello spazio-tempo, abbiamo una regione che si espande nel tempo, contenente tutte le traiettorie possibili compatibili con l'informazione corrente. Il confine di questa regione rappresenta i limiti fisici o logici sulle possibili evoluzioni: un'auto non può superare una certa velocità massima, un pedone non può attraversare pareti solide.

In un framework qualitativo, l'incertezza può essere rappresentata come un insieme di segmenti qualitativi possibili. Per esempio, sappiamo che un oggetto sta attualmente muovendosi verso nord, ma tra un minuto potrebbe essere ancora in movimento verso nord, potrebbe essersi fermato, o potrebbe aver girato verso nord-est o nord-ovest. Questo genera un albero di possibilità future, dove ogni ramo rappresenta uno scenario possibile.

La probability distribution over trajectories può essere modellata usando modelli generativi stocastici come Hidden Markov Models (HMM) o processi Gaussiani. In questi modelli, lo stato latente rappresenta proprietà cinematiche non direttamente osservabili (come l'intenzione di movimento), e le osservazioni sono le posizioni effettive dell'oggetto nel

tempo. L'inferenza bayesiana permette di aggiornare la distribuzione di probabilità sulle traiettorie future man mano che nuove osservazioni diventano disponibili.

Integrazione con Ragionamento Temporale

Le logiche cinematiche sono naturalmente integrate con logiche temporali, perché il movimento è un fenomeno intrinsecamente temporale. L'integrazione permette di esprimere proprietà complesse che combinano aspetti spaziali e temporali.

Interval Temporal Logic per Configurazioni Cinematiche

L'Interval Temporal Logic di Allen fornisce un framework naturale per ragionare su configurazioni cinematiche che persistono per intervalli di tempo. Una configurazione QTC, per esempio, può essere vista come un fatto che è vero durante un certo intervallo temporale. Le tredici relazioni di Allen (before, meets, overlaps, during, starts, finishes, e le loro inverse, più equals) possono essere utilizzate per esprimere vincoli temporali tra intervalli durante i quali diverse configurazioni cinematiche sono valide.

Per esempio, possiamo esprimere che la configurazione dove A si avvicina a B precede (before) la configurazione dove A e B sono stazionari uno rispetto all'altro, che a sua volta precede la configurazione dove A si allontana da B. Questo descrive un pattern di avvicinamento, incontro e allontanamento.

La composizione di relazioni temporali di Allen si combina con la composizione di relazioni cinematiche per produrre un calcolo spazio-temporale integrato. Date relazioni temporali tra intervalli I_1 , I_2 e I_3 , e relazioni cinematiche valide durante questi intervalli, possiamo inferire vincoli sulle possibili evoluzioni del sistema.

Event Calculus per Dinamiche di Movimento

L'Event Calculus è un formalismo logico per ragionare su eventi e cambiamenti di stato nel tempo. Può essere utilizzato per modellare le dinamiche del movimento, dove gli eventi sono azioni che cambiano lo stato cinematico (accelerazione, sterzata, fermata), e i fluenti sono proprietà che cambiano nel tempo (posizione, velocità, configurazione cinematica).

Gli assiomi dell'Event Calculus specificano come gli eventi iniziano o terminano la validità di fluenti. Per esempio, un evento "iniziaAccelerazione(oggetto)" causa l'inizio della validità del fluente "staAccelerando(oggetto)", che rimane vero finché non si verifica un evento "terminaAccelerazione(oggetto)" o "iniziaDecelerazione(oggetto)".

Le precondizioni e gli effetti degli eventi possono includere vincoli spaziali. Per esempio, l'evento "svolta(oggetto, sinistra)" può avere come precondizione che l'oggetto sia in movimento, e come effetto il cambiamento della direzione del movimento verso sinistra rispetto alla direzione corrente.

Il reasoning in Event Calculus permette di rispondere a query come: "dato questo scenario iniziale e questa sequenza di eventi, quali configurazioni cinematiche saranno vere al tempo

$t?$ " oppure "quali sequenze di eventi possono portare dalla configurazione iniziale alla configurazione goal?". Questi sono problemi di simulation e planning rispettivamente.

Temporal Constraint Networks per Sincronizzazione

Quando ragioniamo su molteplici oggetti in movimento, spesso dobbiamo gestire vincoli temporali sulla sincronizzazione dei loro movimenti. Per esempio, due robot che devono incontrarsi in un punto specifico devono coordinare i loro tempi di arrivo. Le Temporal Constraint Networks (TCN) forniscono una rappresentazione efficiente per questo tipo di vincoli.

Una TCN è un grafo dove i nodi rappresentano time points (istanti di tempo o eventi temporali) e gli archi rappresentano vincoli sulla differenza temporale tra i nodi. Un vincolo può essere un intervallo: l'evento B deve verificarsi tra cinque e dieci secondi dopo l'evento A. L'algoritmo di Floyd-Warshall può essere utilizzato per propagare questi vincoli e determinare se la rete è consistente (ammette almeno una soluzione).

Nel contesto cinematografico, i time points possono rappresentare eventi come "A raggiunge la posizione P" o "B inizia a decelerare". I vincoli temporali tra questi eventi derivano dalle leggi fisiche del movimento e dai requisiti della coordinazione. Per esempio, se A e B devono incontrarsi in P, e A è attualmente più lontano da P di quanto lo sia B, allora A deve iniziare a muoversi prima di B, o muoversi più velocemente, per arrivare contemporaneamente.

Constraint-Based Spatial Logics

Fondamenti dei Vincoli Spaziali

Le logiche basate su vincoli rappresentano un paradigma potente e versatile per la rappresentazione e il ragionamento spaziale. L'idea centrale è quella di descrivere configurazioni spaziali non attraverso asserzioni logiche tradizionali, ma attraverso sistemi di vincoli matematici che le configurazioni devono soddisfare. Questo approccio è particolarmente naturale per lo spazio, perché molte proprietà geometriche e relazioni spaziali possono essere espresse elegantemente come equazioni o disequazioni algebriche.

Un vincolo spaziale è una formula matematica che coinvolge variabili spaziali e che deve essere soddisfatta affinché una configurazione sia considerata valida. Le variabili spaziali possono rappresentare punti, coordinate, distanze, angoli, aree, o altri parametri geometrici. I vincoli possono essere equazioni (uguaglianze), disequazioni (disuguaglianze), o formule logiche che combinano multiple condizioni algebriche.

Per esempio, il vincolo che tre punti p_1, p_2, p_3 siano collineari può essere espresso come: $\det([p_2 - p_1, p_3 - p_1]) = 0$, dove $\det$ denota il determinante della matrice formata dai vettori. Il vincolo che un punto p sia all'interno di un cerchio di centro c e raggio r è: $\|p - c\|^2 \leq r^2$. Il vincolo che due segmenti di linea non si intersechino può essere espresso come un sistema di disequazioni che verificano che i segmenti giacciono su lati opposti l'uno dell'altro.

Il potere delle logiche basate su vincoli deriva da diversi fattori. Primo, esse forniscono un linguaggio dichiarativo e naturale per esprimere requisiti geometrici complessi senza dover specificare algoritmi procedurali per verificarli o soddisfarli. Secondo, tecniche algoritmiche sofisticate di constraint satisfaction possono essere applicate per trovare configurazioni che soddisfano i vincoli, verificare consistenza, o derivare nuovi vincoli per propagazione. Terzo, l'approccio si integra naturalmente con metodi di ottimizzazione, permettendo non solo di trovare configurazioni valide ma di trovare configurazioni ottimali secondo qualche criterio obiettivo.

Constraint Databases con Geometria Semi-Algebrica

I Constraint Databases rappresentano un framework teorico e pratico per database che memorizzano e interrogano oggetti spaziali infiniti definiti attraverso vincoli. Questo contrasta con i database spaziali tradizionali che memorizzano rappresentazioni esplicite finite di oggetti (collezioni di vertici, collezioni di pixel, decomposizioni cellulari finite).

Modello dei Constraint Databases

Nel modello dei constraint databases, un oggetto spaziale è rappresentato da una formula logica del primo ordine i cui modelli sono esattamente i punti che appartengono all'oggetto. Per esempio, un rettangolo può essere rappresentato dalla formula: $\psi_{\text{rect}}(x, y) \equiv (a \leq x \leq b) \wedge (c \leq y \leq d)$, dove a, b, c, d sono costanti che definiscono i limiti del rettangolo. Un punto (x_0, y_0) appartiene al rettangolo se e solo se $\psi_{\text{rect}}(x_0, y_0)$ è vera.

Questo approccio si estende a geometrie arbitrariamente complesse. Un cerchio: $\psi_{\text{circle}}(x, y) \equiv (x - c_x)^2 + (y - c_y)^2 \leq r^2$. Un poligono può essere rappresentato come congiunzione di disequazioni lineari, una per ogni lato: $\psi_{\text{polygon}}(x, y) \equiv \bigwedge_i (a_i x + b_i y + c_i \geq 0)$. Un'unione di regioni può essere rappresentata da una disgiunzione di formule.

La classe di formule tipicamente considerata nei constraint databases è quella delle formule della geometria semi-algebrica reale: formule del primo ordine costruite da variabili reali, costanti numeriche, operazioni aritmetiche (+, -, ×), relazioni di confronto (<, ≤, =, ≥, >), e connettivi logici (∧, ∨, ¬, ∃, ∀). Per il teorema di eliminazione dei quantificatori di Tarski, ogni formula di questo linguaggio con quantificatori può essere trasformata in una formula equivalente senza quantificatori (una combinazione booleana di disequazioni polinomiali).

Questo significa che gli oggetti rappresentabili nei constraint databases sono esattamente gli insiemi semi-algebrici: insiemi definiti da disequazioni polinomiali. La classe degli insiemi semi-algebrici è estremamente ricca: include tutti i poligoni, cerchi, ellissi, curve algebriche, superfici quadriche, e innumerevoli altre forme. Inoltre, gli insiemi semi-algebrici sono chiusi rispetto alle operazioni booleane (unione, intersezione, complemento) e alle proiezioni, il che è fondamentale per le query sui database.

Query su Constraint Databases

Le query su constraint databases sono anch'esse espresse come formule logiche. Una query può essere vista come una funzione che prende uno stato del database (un insieme di

relazioni, ciascuna definita da una formula) e produce un risultato (anch'esso una relazione definita da una formula).

Le query di base nel modello relazionale (selezione, proiezione, join) si traducono naturalmente in operazioni su formule. La selezione di elementi che soddisfano un predicato θ da una relazione definita da φ produce una nuova relazione definita da $\varphi \wedge \theta$. La proiezione di una relazione su un sottoinsieme di attributi corrisponde all'eliminazione dei quantificatori: se R è definita da $\varphi(x, y, z)$ e vogliamo proiettare su (x, y) , il risultato è definito da $\exists z \varphi(x, y, z)$.

Query spaziali più complesse sono altrettanto esprimibili. La query "trovare tutti i punti che sono all'interno della regione A e a distanza minore di d dalla regione B " può essere espressa come: $\psi_{\text{query}}(x, y) \equiv \psi_A(x, y) \wedge \exists x', y' (\psi_B(x', y') \wedge (x - x')^2 + (y - y')^2 \leq d^2)$. Dopo l'eliminazione dei quantificatori, questa formula può essere semplificata in una formula senza quantificatori che definisce esplicitamente il risultato.

Un aspetto potente è che le query sui constraint databases possono produrre come risultati oggetti geometrici infiniti complessi che non erano esplicitamente memorizzati nel database. Per esempio, il buffer di una linea (la regione di tutti i punti a distanza minore di d dalla linea) è un oggetto bidimensionale derivato da un oggetto unidimensionale. I constraint databases rappresentano naturalmente questo tipo di trasformazioni senza dover ricorrere a approssimazioni o discretizzazioni.

Complessità Computazionale e Implementazioni

La teoria dei constraint databases fornisce eleganza concettuale e potere espressivo, ma solleva questioni di efficienza computazionale. L'eliminazione dei quantificatori in geometria semi-algebrica reale è computazionalmente costosa: nel caso peggiore, la complessità è doubly exponential nel numero di quantificatori alternanti. Inoltre, la dimensione delle formule (numero di termini polinomiali) può crescere esponenzialmente durante l'eliminazione.

Nella pratica, diverse strategie rendono i constraint databases utilizzabili. Primo, molte query pratiche hanno struttura semplice che permette eliminazione efficiente. Query che coinvolgono solo vincoli lineari (geometria convessa, poliedri) possono essere processate in tempo polinomiale usando programmazione lineare e algoritmi geometrici. Secondo, tecniche di lazy evaluation evitano di calcolare completamente il risultato di query intermedie, mantenendole in forma simbolica e valutandole solo quando necessario. Terzo, approssimazioni conservative possono essere utilizzate: invece di calcolare esattamente la regione risultato, calcoliamo una sovra-approssimazione semplice che contiene certamente il risultato esatto.

Implementazioni pratiche di constraint databases includono il sistema DEDALE, che si concentra su vincoli lineari, e sistemi basati su Computer Algebra Systems come Mathematica o Maple per gestire vincoli polinomiali più generali. L'integrazione con database relazionali tradizionali può essere realizzata attraverso extension types che rappresentano oggetti constraint-based come nuovi tipi di dati gestiti da librerie specializzate.

Vincoli Lineari e Geometria Convessa

Una sottoclasse particolarmente importante e trattabile dei vincoli spaziali è quella dei vincoli lineari: disequazioni della forma $a_1x_1 + a_2x_2 + \dots + a_nx_n + b \geq 0$, dove a_i e b sono costanti reali e x_i sono variabili. I vincoli lineari definiscono regioni convesse nello spazio, e la loro intersezione (congiunzione) è ancora una regione convessa, tipicamente un poliedro convesso.

Poliedri Convessi e Half-Spaces

Un half-space in n dimensioni è l'insieme di punti che soddisfano una singola disequazione lineare: $H = \{x \in \mathbb{R}^n : a \cdot x + b \geq 0\}$, dove a è un vettore normale e b è un offset scalare. Geometricamente, un half-space è tutto lo spazio su un lato di un iperpiano.

Un poliedro convesso è l'intersezione di un numero finito di half-spaces: $P = \bigcap_{i=1}^m H_i = \{x : Ax + b \geq 0\}$, dove A è una matrice $m \times n$ e b è un vettore m -dimensionale. Alternativamente, un poliedro può essere rappresentato come la convex hull (involuppo convesso) di un insieme finito di punti vertici, o come la somma di Minkowski di un politopo (poliedro limitato) e un cono poliedrale (insieme di raggi).

La teoria della programmazione lineare fornisce algoritmi efficienti per operazioni su poliedri convessi. Dato un sistema di disequazioni lineari, possiamo verificare in tempo polinomiale se l'insieme soluzione è vuoto (il sistema è inconsistente), trovare un punto nella soluzione, o ottimizzare una funzione obiettivo lineare sulla regione soluzione. Algoritmi come il simplesso o metodi di interior-point sono strumenti fondamentali.

Reasoning su Configurazioni Poliedrali

Molti problemi di ragionamento spaziale possono essere formulati come problemi di soddisfacimento di vincoli lineari. Per esempio, il problema di verificare se un punto p è all'interno di un poliedro P richiede semplicemente di verificare se p soddisfa tutte le disequazioni che definiscono P . Il problema di verificare se due poliedri P_1 e P_2 si intersecano può essere formulato come il problema di trovare un punto x che soddisfa sia i vincoli di P_1 che quelli di P_2 : se tale punto esiste, i poliedri si intersecano, altrimenti sono disgiunti.

Il problema di calcolare l'intersezione di poliedri è diretto: l'intersezione $P_1 \cap P_2$ è semplicemente definita dall'unione dei vincoli: se $P_1 = \{x : A_1x \geq b_1\}$ e $P_2 = \{x : A_2x \geq b_2\}$, allora $P_1 \cap P_2 = \{x : A_1x \geq b_1 \wedge A_2x \geq b_2\}$. Questo è computazionalmente economico, richiede solo di concatenare le liste di disequazioni.

Il problema di calcolare l'unione di poliedri è più complesso, perché l'unione di poliedri convessi non è in generale convessa. Tuttavia, l'unione può essere rappresentata come unione di poliedri convessi (in forma normale disgiuntiva di vincoli), oppure usando strutture dati più complesse che mantengono l'unione implicitamente.

La proiezione di un poliedro su un sottospazio è un'operazione fondamentale corrispondente all'eliminazione di variabili. Data $P = \{(x, y) : A(x, y) \geq b\}$, la proiezione su x è: $\text{proj}_x(P) = \{x : \exists y (A(x, y) \geq b)\}$. Il teorema di Fourier-Motzkin fornisce un algoritmo per l'eliminazione di

quantificatori lineari: per eliminare una variabile y , combiniamo opportunamente le disequazioni per derivare nuove disequazioni che coinvolgono solo le variabili rimanenti. Tuttavia, il numero di disequazioni può crescere esponenzialmente nel caso peggiore.

Constraint Propagation per Vincoli Lineari

La propagazione di vincoli è una tecnica fondamentale per risolvere problemi di soddisfacimento di vincoli. L'idea è quella di iterativamente restringere i domini delle variabili eliminando valori che non possono partecipare ad alcuna soluzione globale, finché non si raggiunge un punto fisso dove ulteriori restrizioni non sono possibili, oppure si scopre che il sistema è inconsistente (il dominio di qualche variabile diventa vuoto).

Per vincoli lineari su variabili continue, la propagazione opera restringendo gli intervalli di valori possibili per ciascuna variabile. Data una disequazione lineare $a_1x_1 + \dots + a_nx_n + b \geq 0$ e intervalli correnti $[L_i, U_i]$ per ciascuna variabile x_i , possiamo derivare nuovi bound su una variabile, diciamo x_j , considerando i valori estremi delle altre variabili che renderebbero la disequazione più difficile da soddisfare.

Formalmente, risolvendo per x_j , otteniamo: $x_j \geq (b - \sum_{i \neq j} a_i x_i) / a_j$ (assumendo $a_j > 0$). Per minimizzare il lato destro (ottenere il bound più stretto su x_j), scegliamo x_i al loro estremo che minimizza $a_i x_i$: se $a_i > 0$ scegliamo $x_i = L_i$, se $a_i < 0$ scegliamo $x_i = U_i$. Questo produce un nuovo lower bound per x_j . Ripetiamo per tutte le variabili e tutte le disequazioni fino a convergenza.

Questa tecnica, nota come bounds consistency o 2B-consistency per vincoli lineari, non garantisce di trovare tutte le inconsistenze (non è completa), ma è computazionalmente efficiente e spesso sufficiente in pratica. Per completezza, tecniche più costose come simplex-based consistency checking o branch-and-bound possono essere utilizzate.

Vincoli Polinomiali e Geometria Algebrica

Estendendo oltre i vincoli lineari, i vincoli polinomiali permettono di rappresentare una classe molto più ampia di regioni e relazioni spaziali. Un vincolo polinomiale ha la forma $p(x_1, \dots, x_n) \circ 0$, dove p è un polinomio multivariato con coefficienti reali, e $\circ$ è una relazione di confronto ($<$, $\leq$, $=$, $\geq$, $>$).

Varietà Algebriche e Insiemi Semi-Algebrici

Una varietà algebrica è l'insieme di zeri di un sistema di equazioni polinomiali: $V = \{x : p_1(x) = 0 \wedge \dots \wedge p_m(x) = 0\}$. Geometricamente, ogni equazione polinomiale definisce una ipersuperficie nello spazio, e la varietà è l'intersezione di queste ipersuperfici. Per esempio, l'equazione $x^2 + y^2 - 1 = 0$ definisce un cerchio nel piano, l'equazione $x^2 + y^2 + z^2 - 1 = 0$ definisce una sfera nello spazio tridimensionale.

Un insieme semi-algebrico è un insieme definito da una combinazione booleana di equazioni e disequazioni polinomiali: $S = \{x : \phi(x)\}$, dove ϕ è una formula costruita da atomi della forma $p(x) \circ 0$ usando $\wedge$, $\vee$, $\neg$. Gli insiemi semi-algebrici generalizzano le varietà algebriche permettendo disequazioni e complementi. Per esempio, l'interno di un cerchio è un insieme

semi-algebrico ($x^2 + y^2 < 1$), così come l'esterno di un'ellisse, o l'unione di molteplici regioni polinomialmente definite.

Gli insiemi semi-algebrici hanno proprietà matematiche notevoli. Sono chiusi rispetto a tutte le operazioni booleane (unione, intersezione, complemento) e rispetto a proiezioni. Ogni insieme semi-algebrico ha un numero finito di componenti connesse. Ogni insieme semi-algebrico può essere decomposto in una unione finita di manifolds smooth. Queste proprietà rendono gli insiemi semi-algebrici una classe naturale per la rappresentazione di geometrie complesse.

Cylindrical Algebraic Decomposition

Un algoritmo fondamentale per il reasoning su vincoli polinomiali è la Cylindrical Algebraic Decomposition (CAD), sviluppata da George Collins negli anni Settanta. La CAD scompone lo spazio $\mathbb{R}^n$ in una unione finita di celle cilindricamente disposte, tali che su ciascuna cella, ogni polinomio in un insieme dato di polinomiali ha segno costante (sempre positivo, sempre negativo, o sempre zero).

La costruzione della CAD procede induttivamente sulla dimensione. Per $n = 1$ (la retta reale), identifichiamo le radici reali di tutti i polinomiali in input. Queste radici dividono la retta in intervalli e punti. Su ogni intervallo, tutti i polinomiali hanno segno costante. Per $n > 1$, proiettiamo i polinomiali su $n-1$ dimensioni, costruiamo ricorsivamente la CAD dello spazio proiettato, e poi per ogni cella nella CAD proiettata, "solleviamo" la cella allo spazio n -dimensionale identificando le sezioni e i settori definiti dai polinomiali sopra quella cella.

Il risultato è una decomposizione dove ogni cella è definita da vincoli della forma " x_{n-1} sta in una certa cella della CAD di $\mathbb{R}^{n-1}$, e x_n soddisfa certi vincoli (o è uguale a una certa funzione algebrica di $x_1, \dots, x_{n-1}$, oppure sta in un intervallo tra tali funzioni)". Questa struttura cilindrica garantisce che le celle si impilino ordinatamente.

Data una CAD, rispondere a query su formule polinomiali diventa semplice: per ogni cella, valutiamo il segno di ciascun polinomiale (che è costante sulla cella), e quindi valutiamo la formula booleana. Le celle dove la formula è vera costituiscono la regione soluzione, e possiamo facilmente contare componenti connesse, calcolare dimensioni, estrarre punti campione, e così via.

La limitazione principale della CAD è la complessità computazionale: la dimensione della decomposizione e il tempo di calcolo sono doubly exponential nel numero di variabili nel caso peggiore. Per problemi pratici in due o tre dimensioni con pochi polinomiali, la CAD è spesso fattibile, ma per dimensioni alte o molti polinomiali, diventa rapidamente intrattabile. Ricerca corrente si concentra su varianti della CAD che evitano decomposizioni complete quando non necessarie, costruendo solo le parti rilevanti per rispondere a una specifica query.

Satisfiability Modulo Theories (SMT) per Vincoli Non-Lineari

Un approccio moderno al reasoning su vincoli polinomiali è attraverso SMT solvers (Satisfiability Modulo Theories). Gli SMT solvers combinano SAT solvers (per la struttura

booleana di formule) con theory solvers specializzati (per vincoli aritmetici, vincoli sugli array, vincoli su stringhe, ecc.).

Per vincoli polinomiali reali, la theory di interesse è Real Closed Fields (RCF) o Non-Linear Real Arithmetic (NRA). Gli SMT solvers per NRA utilizzano tecniche come CAD, virtual substitution, o interval constraint propagation per decidere soddisfacibilità di congiunzioni di vincoli polinomiali, e le combinano con euristiche SAT per gestire la struttura disgiuntiva di formule complesse.

Un vantaggio degli SMT solvers è che sono progettati per restituire rapidamente "unsat" (insoddisfacibile) quando una formula è inconsistente, spesso senza dover costruire una CAD completa. Possono anche produrre modelli (assegnamenti alle variabili che soddisfano i vincoli) quando una formula è soddisfacibile, e certificati di insoddisfacibilità quando non lo è.

Strumenti come Z3, CVC5, Yices, e MathSAT includono solver per NRA e sono utilizzati in verifica formale, bounded model checking, test generation, e molte altre applicazioni dove è necessario ragionare su vincoli geometrici complessi.

Query Languages Spaziali Basate su Vincoli

L'approccio basato su vincoli permette di progettare query languages spaziali dichiarativi molto espressivi. Invece di specificare algoritmi procedurali per calcolare risultati di query, l'utente specifica vincoli che i risultati devono soddisfare, e il sistema è responsabile di trovare configurazioni soddisfacenti.

SQL Esteso con Vincoli Spaziali

Un'estensione naturale di SQL per supportare query spaziali basate su vincoli è quella di permettere condizioni WHERE che includono vincoli geometrici. Per esempio:

```
SELECT building.name, park.name FROM buildings AS building, parks AS park WHERE
Distance(building.location, park.boundary) < 500 AND building.height > 20
```

Questa query trova coppie di edifici e parchi dove l'edificio è alto più di 20 metri e dista meno di 500 metri dal confine del parco. La funzione Distance e le geometrie (location, boundary) sono trattate attraverso un solver di vincoli geometrici.

Più potentemente, possiamo avere variabili spaziali non istanziate in input, e chiedere al sistema di trovare valori per esse. Per esempio:

```
SELECT x, y FROM terrain WHERE terrain.elevation(x, y) > 1000 AND EXISTS building IN
buildings (Distance((x, y), building.location) < 200)
```

Questa query cerca punti (x, y) che sono ad altitudine maggiore di 1000 metri e distano meno di 200 metri da qualche edificio. Il sistema deve risolvere i vincoli quantificati per trovare la regione soluzione.

Linguaggi Dichiarativi per Constraint Specification

Linguaggi specializzati come ConGolog (Constraint Golog) permettono di specificare non solo query ma anche azioni e piani che rispettano vincoli spaziali. Un programma ConGolog può includere azioni che modificano lo stato spaziale (muovere un oggetto, costruire una struttura) e vincoli che devono essere mantenuti invarianti (nessun oggetto si sovrappone, tutti gli oggetti sono connessi, il centro di massa rimane in una certa regione).

L'esecutore del programma ConGolog utilizza constraint solving per verificare che ciascuna azione proposta sia fattibile (non viola vincoli) e per trovare parametri delle azioni che soddisfano obiettivi desiderati. Questo è utile in pianificazione robotica, progettazione assistita, e molte altre applicazioni.

Un esempio di statement ConGolog potrebbe essere:

```
place_object(object, container):
  precondition: fitsInside(object.size, container.volume)
  effect: contains(container, object)
  constraint:  $\forall$  other IN container:  $\neg$ overlaps(object.geometry, other.geometry)
```

Questo specifica che l'azione di piazzare un oggetto in un contenitore richiede che l'oggetto ci stia (vincolo di volume), e che dopo l'azione, nessun oggetto nel contenitore si sovrapponga agli altri (vincolo geometrico). Il sistema di planning deve trovare una posizione e orientamento per l'oggetto che soddisfi il vincolo di non-sovrapposizione.

Integrazione con Logiche di Ordine Superiore

Per applicazioni molto espressive, i vincoli spaziali possono essere integrati in logiche di ordine superiore dove quantifichiamo non solo su punti o oggetti, ma su proprietà, regioni, o funzioni. Per esempio, potremmo voler esprimere: "esiste una regione R tale che R è connessa, R ha area maggiore di 100, e ogni punto in R è a distanza minore di 10 da qualche albero".

Questo richiede quantificazione su insiemi di punti (la regione R). Formalmente: $\exists R \subseteq \text{Space} : \text{connected}(R) \wedge \text{area}(R) > 100 \wedge \forall p \in R \exists t \in \text{Trees} : \text{distance}(p, t) < 10$.

Il reasoning in queste logiche di ordine superiore è indecidibile in generale, ma frammenti decidibili possono essere identificati imponendo restrizioni sulla struttura delle formule o sulla classe di oggetti su cui quantifichiamo. Per esempio, quantificazione su regioni poligonali con un numero limitato di vertici, o quantificazione su regioni semi-algebriche di grado polinomiale limitato.

Ottimizzazione con Vincoli Spaziali

Oltre alla pura soddisfacibilità, molti problemi pratici richiedono di trovare non solo configurazioni che soddisfano vincoli, ma configurazioni ottimali secondo qualche criterio.

L'ottimizzazione con vincoli combina tecniche di constraint satisfaction con tecniche di ottimizzazione matematica.

Programmazione Matematica con Vincoli Geometrici

La programmazione matematica standard (programmazione lineare, programmazione non-lineare, programmazione intera) può essere estesa con vincoli geometrici specifici del dominio spaziale. Per esempio, il problema di facility location chiede di posizionare un certo numero di strutture (ospedali, magazzini, stazioni di servizio) in modo da ottimizzare qualche obiettivo (minimizzare la distanza massima da qualsiasi punto della regione servita, massimizzare la copertura con budget limitato) rispettando vincoli spaziali (le strutture non possono essere posizionate in zone proibite, devono essere connesse da strade esistenti).

Formalmente, questo può essere modellato come:

minimize: $f(x_1, \dots, x_n)$
 soggetto a: $g_i(x_1, \dots, x_n) \leq 0$ per $i = 1, \dots, m$
 $h_j(x_1, \dots, x_n) = 0$ per $j = 1, \dots, p$
 $x \in S$

dove x_k sono le coordinate delle strutture da posizionare, f è la funzione obiettivo, g_i e h_j sono vincoli funzionali, e S è un vincolo spaziale (per esempio, S potrebbe essere un insieme semi-algebrico rappresentante la regione fattibile).

Algoritmi di ottimizzazione locale (gradient descent, metodi di punto interiore) possono essere utilizzati se le funzioni sono differenziabili. Algoritmi di ottimizzazione globale (branch and bound, algoritmi genetici, simulated annealing) possono essere necessari se la funzione obiettivo ha molteplici ottimi locali.

Constraint Optimization Problems (COP) Spaziali

Un Constraint Optimization Problem è la combinazione di un Constraint Satisfaction Problem (trova un assegnamento che soddisfa i vincoli) con una funzione obiettivo da ottimizzare. Nel contesto spaziale, le variabili possono rappresentare coordinate, orientamenti, forme, e i vincoli codificano requisiti geometrici.

Tecniche specializzate per COP includono branch-and-bound con propagazione di vincoli: l'albero di ricerca esplora assegnamenti parziali alle variabili, la propagazione di vincoli restringe i domini delle variabili non ancora assegnate, e bound sull'obiettivo permettono di potare rami dell'albero che non possono contenere soluzioni migliori della migliore soluzione trovata finora.

Per problemi spaziali continui, tecniche di decomposizione possono discretizzare lo spazio in una griglia e risolvere un problema discreto approssimato, poi raffinare la soluzione ottimizzando localmente intorno alla soluzione discreta. Alternativamente, metodi di campionamento come Monte Carlo Tree Search possono esplorare stocasticamente lo spazio continuo, concentrando i campioni in regioni promettenti.

Multi-Objective Optimization Spaziale

Molti problemi pratici hanno obiettivi multipli conflittuali. Per esempio, nel posizionare edifici in un piano urbano, potremmo voler massimizzare la densità (utilizzare efficientemente lo spazio), massimizzare l'accesso alla luce solare (evitare ombreggiature), minimizzare i costi di costruzione, e massimizzare l'estetica della configurazione.

L'ottimizzazione multi-obiettivo non produce una singola soluzione ottima, ma un insieme di soluzioni Pareto-ottimali: soluzioni tali che non è possibile migliorare un obiettivo senza peggiorare almeno un altro. Formalmente, una soluzione x domina una soluzione y ($x \succ y$) se x è almeno buona come y su tutti gli obiettivi e strettamente migliore su almeno un obiettivo. L'insieme di soluzioni non dominate costituisce il Pareto front.

Algoritmi evolutivi come NSGA-II (Non-dominated Sorting Genetic Algorithm) sono ampiamente utilizzati per problemi di ottimizzazione spaziale multi-obiettivo. Questi algoritmi mantengono una popolazione di soluzioni candidate, le evolvono attraverso operatori genetici (crossover, mutazione), e selezionano per la prossima generazione un insieme diversificato di soluzioni non dominate.

Nel contesto spaziale, gli operatori genetici devono essere progettati per rispettare vincoli geometrici. Per esempio, se le soluzioni sono configurazioni di edifici, l'operatore di crossover potrebbe combinare sotto-regioni di configurazioni parent, e l'operatore di mutazione potrebbe spostare, ruotare, o ridimensionare edifici individuali. meccanismi di riparazione possono correggere configurazioni che violano vincoli duri (per esempio, spostando leggermente un edificio che interseca un altro, o ridimensionando una regione che eccede i limiti consentiti), ma la difficoltà principale rimane quella di mantenere la coerenza geometrica delle soluzioni durante l'intero processo evolutivo. In un contesto spaziale continuo, infatti, ogni operatore genetico deve preservare la struttura semi-algebrica delle regioni: un poligono rappresentato da vincoli lineari $a_i x + b_i y + c_i \geq 0$ non può essere trasformato in una forma arbitraria senza violare la rappresentazione vincolare; analogamente, una regione definita da un vincolo polinomiale $p(x_1, \dots, x_n) \leq 0$ deve rimanere all'interno della stessa classe semi-algebrica per poter essere manipolata dal solver.

La presenza di obiettivi multipli introduce inoltre una complessità geometrica aggiuntiva: la frontiera di Pareto non è, in generale, un insieme convesso, né può essere descritta da una singola funzione obiettivo aggregata. Ogni soluzione x è valutata da un vettore di funzioni $f(x) = (f_1(x), \dots, f_k(x))$, e la relazione di dominanza $x \succ y$ (cioè: $f_i(x) \leq f_i(y)$ per tutti gli obiettivi i e $f_j(x) < f_j(y)$ per almeno un j) induce una struttura parziale sull'insieme delle configurazioni spaziali. La geometria di questa struttura è spesso intricata: regioni di soluzioni non dominate possono essere separate da vincoli non lineari, possono presentare discontinuità, o possono essere immerse in spazi di dimensione elevata quando le variabili rappresentano coordinate, orientamenti, parametri di forma o funzioni di densità.

Dal punto di vista algoritmico, la ricerca del Pareto front richiede una combinazione di esplorazione globale e raffinamento locale. Gli algoritmi evolutivi forniscono una copertura ampia dello spazio delle soluzioni, ma devono essere integrati con tecniche di ottimizzazione

locale che rispettino i vincoli geometrici, come metodi di proiezione su insiemi ammissibili o gradient-based refinement quando le funzioni obiettivo sono differenziabili. In presenza di vincoli polinomiali $p_j(x) \geq 0$, la proiezione di un punto non ammissibile sulla regione ammissibile può richiedere la risoluzione di un problema di ottimizzazione non lineare, mentre per vincoli lineari $Ax \geq b$ la proiezione può essere calcolata tramite metodi di programmazione quadratica. La combinazione di questi strumenti permette di mantenere la popolazione evolutiva entro regioni geometricamente coerenti, evitando derive numeriche o degenerazioni topologiche.

Un aspetto particolarmente delicato dell'ottimizzazione spaziale multi-obiettivo è la gestione dell'incertezza. Quando gli obiettivi dipendono da parametri incerti θ , la funzione obiettivo diventa una funzione attesa $E_\theta[f_i(x, \theta)]$ o una funzione di rischio come $Pr_\theta[f_i(x, \theta) \leq \tau]$. In questi casi, la frontiera di Pareto non è più definita su uno spazio deterministico, ma su uno spazio di distribuzioni, e la dominanza deve essere interpretata in senso probabilistico. La regione delle soluzioni non dominate può diventare altamente non regolare, e la sua approssimazione richiede tecniche di campionamento stratificato, surrogate models o decomposizioni adattive dello spazio dei parametri.

In definitiva, l'ottimizzazione spaziale multi-obiettivo rappresenta una sintesi naturale ma estremamente sofisticata delle tecniche discusse nelle sezioni precedenti: vincoli lineari e polinomiali, geometria semi-algebrica, propagazione dei vincoli, ragionamento sotto incertezza e metodi di ricerca globale. La sua difficoltà non risiede soltanto nella complessità computazionale, ma nella necessità di mantenere una rappresentazione coerente e manipolabile delle regioni spaziali durante l'intero processo di ottimizzazione. La frontiera di Pareto risultante non è soltanto un insieme di soluzioni ottimali, ma una struttura geometrica che riflette l'interazione profonda tra vincoli, obiettivi e incertezza, e che costituisce un oggetto matematico di grande interesse sia teorico che applicativo. Quando si estendono le logiche spaziali basate su vincoli a contesti in cui incertezza, dinamica temporale e ottimizzazione interagiscono, la nozione stessa di vincolo deve essere reinterpretata. Un vincolo non è più soltanto una relazione deterministica del tipo $g(x_1, \dots, x_n) \geq 0$, ma diventa un oggetto parametrico, spesso dipendente da variabili non osservabili o affette da rumore. Se un vincolo assume la forma $g(x, \theta) \geq 0$, dove θ è un parametro incerto, esso non definisce una singola regione ammissibile, bensì una famiglia di regioni $\{x : g(x, \theta) \geq 0\}$ indicizzata da tutti i possibili valori di θ . La domanda fondamentale diventa allora come interpretare la soddisfaccibilità: se si richiede che il vincolo sia valido per ogni θ in un insieme di incertezza Θ , si ottiene la nozione di robust feasibility, formalmente $\forall \theta \in \Theta : g(x, \theta) \geq 0$. Se invece si accetta una violazione controllata, si entra nel dominio dei chance constraints, dove la condizione diventa $Pr_\theta[g(x, \theta) \geq 0] \geq p$. In questo caso, la regione ammissibile non è più descrivibile da semplici disequazioni polinomiali, ma dipende dalla distribuzione di θ e dalla trasformazione indotta da g . La propagazione dell'incertezza attraverso vincoli polinomiali $p(x_1, \dots, x_n)$ è particolarmente delicata. L'immagine di un insieme incerto X attraverso un polinomio può generare regioni non convesse, con frontiere complesse e dipendenze sensibili dai parametri. Per mantenere una rappresentazione computabile, si ricorre a modelli come i Taylor models, che approssimano un polinomio tramite un'espansione locale $p(x) \approx T(x) + R(x)$ dove $T(x)$ è un polinomio troncato e $R(x)$ un intervallo che garantisce rigore formale. Altre tecniche, come le rappresentazioni di Bernstein, permettono di ottenere bound rigorosi su $p(x)$ su domini rettangolari, sfruttando la

proprietà che i coefficienti di Bernstein delimitano l'intervallo dei valori del polinomio. Questi strumenti consentono di mantenere vincoli del tipo $p(x) \geq 0$ in forma computabile anche quando l'incertezza rende la regione ammissibile difficile da descrivere esattamente. L'introduzione della dimensione temporale porta a vincoli del tipo $g(x(t), t) \geq 0$, dove $x(t)$ è una traiettoria continua. Qui la soddisfacibilità non riguarda più un singolo punto nello spazio, ma un'intera funzione. Vincoli cinematici come $\|x'(t)\| \leq v_{\max}$ o dinamici come $\|x''(t)\| \leq a_{\max}$ si combinano con vincoli geometrici, generando problemi di soddisfacibilità su spazi funzionali. Le logiche spazio-temporali, come STL (Signal Temporal Logic), permettono di esprimere proprietà globali del tipo $G_{[0,T]} g(x(t)) \geq 0$ (cioè: per ogni t in $[0,T]$, il vincolo è soddisfatto) o proprietà reattive come $F_{[0,T]} h(x(t)) \leq 0$ (cioè: esiste un istante in cui un certo vincolo è soddisfatto). La verifica di tali proprietà richiede tecniche di reachability analysis, che calcolano insiemi del tipo $\text{Reach}(t) = \{x(t) : x(0) \in X_0\}$, oppure metodi di ottimizzazione su traiettorie che discretizzano il tempo e risolvono sequenze di vincoli del tipo $g(x_k, t_k) \geq 0$ per $k = 0, \dots, N$. In questo contesto, l'ottimizzazione spaziale non è più un semplice problema di minimizzazione di una funzione $f(x_1, \dots, x_n)$ soggetta a vincoli statici, ma diventa un problema di ottimizzazione su traiettorie, spesso multi-obiettivo, dove si cerca di minimizzare simultaneamente criteri come lunghezza, energia, rischio, distanza da ostacoli, o tempo di percorrenza. La nozione di Pareto-optimality emerge naturalmente: una traiettoria $x(t)$ domina un'altra $y(t)$ se $f_i(x) \leq f_i(y)$ per tutti gli obiettivi i , e strettamente per almeno uno. La frontiera di Pareto risultante può essere altamente non convessa, soprattutto quando i vincoli includono regioni semi-algebriche definite da polinomi $p_j(x) \geq 0$ o vincoli probabilistici del tipo $\Pr[g_k(x, \theta) \geq 0] \geq p_k$. La complessità di questi problemi rende necessario un approccio architetturale stratificato. Al livello simbolico, le formule logiche che definiscono vincoli e query vengono manipolate tramite trasformazioni sintattiche, eliminazione di quantificatori e normalizzazioni. Al livello geometrico, si opera su poliedri, insiemi semi-algebrici, decomposizioni cilindriche parziali e rappresentazioni conservative dell'incertezza. Al livello numerico, si risolvono sistemi del tipo $Ax \geq b$, $p(x) \geq 0$, $\min f(x)$ tramite metodi di ottimizzazione lineare, non lineare o globale. Al livello probabilistico, si aggiornano distribuzioni $\Pr(\theta)$ e si valutano vincoli del tipo $\Pr[g(x, \theta) \geq 0]$. Infine, al livello decisionale, si integrano i risultati per generare piani, traiettorie o configurazioni ottimali. L'immagine complessiva è quella di un paradigma unificante, in cui logica, geometria, algebra, probabilità e ottimizzazione convergono in un'unica struttura concettuale. Le logiche spaziali basate su vincoli non sono semplicemente un'estensione delle logiche classiche, ma un linguaggio dichiarativo capace di rappresentare configurazioni geometriche complesse, dinamiche temporali, incertezza e obiettivi multipli, mantenendo una sintassi compatta e una semantica rigorosa. La loro forza risiede nella capacità di delegare a solver specializzati la manipolazione di regioni geometriche intricate, la verifica di consistenza, la propagazione dei vincoli e l'ottimizzazione multi-criterio, preservando al contempo un livello di astrazione elevato e una notevole espressività formale.